世界知的所有権機関
国際事務局

PCT

特許協力条約に基づいて公開された国際出願

(51) 国際特許分類6 G11B 20/18, H03M 13/00	A1	(11) 国際公開番号	WO99/63540
		(43) 国際公開日	1999年12月9日(09.12.99)

(21) 国際出願番号 PCT/JP98/02476

(22) 国際出願日 1998年6月4日(04.06.98)

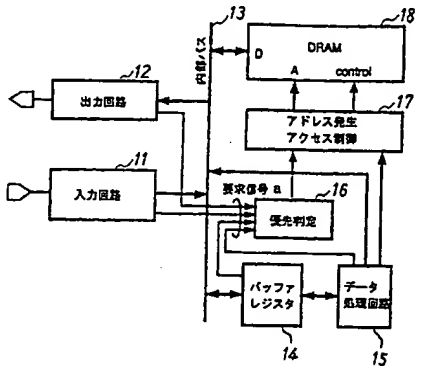
(71) 出願人（米国を除くすべての指定国について）
株式会社 日立製作所(HITACHI, LTD.)[JP/JP]
〒101-8010 東京都千代田区神田駿河台四丁目6番地
Tokyo, (JP)
(72) 発明者；および
(75) 発明者／出願人（米国についてのみ）
小島浩嗣(KOJIMA, Hirotsugu)[JP/JP]
金子憲二(KANEKO, Kenji)[JP/JP]
武井宣幸(TAKEI, Nobuyuki)[JP/JP]
〒187-8588 東京都小平市上水本町五丁目20番1号
株式会社 日立製作所 半導体事業部内 Tokyo, (JP)
(74) 代理人
弁理士 磯村雅俊(ISOMURA, Masatoshi)
〒160-0023 東京都新宿区西新宿1丁目23番1号
新宿千葉ビル Tokyo, (JP)

(81) 指定国 AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, GM, GW, HU, ID, IL, IS, JP, KE, KG, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, 欧州特許 (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI特許 (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), ARIPO特許 (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), ユーラシア特許 (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM)

添付公開書類
国際調査報告書

(54)Title: DATA PROCESSOR AND DATA PROCESSING METHOD

(54)発明の名称 データ処理装置および処理方法



14 ... Buffer register
15 ... Data processing circuit
16 ... Priority discrimination
17 ... Address generation/access control
a ... Request signal
11 ... Input circuit
12 ... Output circuit
13 ... Internal bus

(57) Abstract

A data processor provided with an input circuit means (11) for pre-processing signals read out from a medium, a first DRAM access priority discriminating means (16) for writing pre-processed results in a DRAM, a second DRAM access means (15) for reading out data for second correction from the above-mentioned DRAM, a third DRAM access output circuit means (12) for reading out data for outputting corrected data to a succeeding stage, and an access control circuit (17) constituted so that the first, the second, and the third DRAM accesses may be stored in the same row address over several words. Thus, the data processor does not require any expensive high-speed memory, because the processor can utilize high-speed access in the same row address of the DRAM and can commonly cope with error correction on signals from different media.

(57)要約

媒体から読み出した信号の前処理を行う入力回路手段（１１）と、前処理した結果をＤＲＡＭに書き込む（第１のＤＲＡＭアクセス）優先判定手段（１６）と、前記ＤＲＡＭから第２の訂正のためのデータを読み出す（第２のＤＲＡＭアクセス）手段（１５）と、訂正したデータを後段に出力するためのデータを読み出す（第３のＤＲＡＭアクセス）出力回路手段（１２）と、前記第１、第２及び第３のＤＲＡＭアクセスが数ワードに渡ってそれぞれ同一ＲＯＷアドレス内に収納されるように構成されたアクセス制御回路（１７）を備える。ＤＲＡＭの同一ＲＯＷアドレス内の高速なアクセスを利用でき、異なる媒体からの信号の誤り訂正に共通に対応でき、高速で高価なメモリを用いる必要がない。

DETAILED DESCRIPTION

[Detailed Description of the Invention]
technical field DVD (Digital Versatile Disk) and CD-ROM (Compact Disk Read Only Memory) to which this invention used dynamic random access memory (Following DRAM is called) for the data buffer -- or it is related with the data processor and art containing the read-out equipment for the error correction which can respond to both.
Background technique As a next-generation personal computer peripheral device, DVD drive equipment has attracted attention. It is de facto indispensable requirements that the DVD drive equipment of the form which replaces CD-ROM by which current loading is carried out can deal with the CD-ROM disk used abundantly from the former. When it constitutes the equipment which reproduces the signal read from the disk of CD-ROM and each DVD drive equipment, it is an important technical problem to develop the configuration means which can respond to both. At this time, it is **** in the case of developing the configuration means which can respond to both that record formats of DVD and CD-ROM differ remarkably.
First, the specification of the data format of CD-ROM is briefly explained using drawing 19 . Drawing 19 (a) is the array Fig. of the time series data after a CD-ROM recovery, and drawing 19 (b) is the format Fig. of the error correcting code of CD-ROM data. The signal read from the disk turns into a 8 bits (1 byte) time series signal as shown in drawing 19 (a) through a demodulator circuit. It becomes one section by the frame and the call and 98 frames about 32 bytes of data stream following 1 byte of sub-code, and it. One section consists of 98 frames which begins from the special sub-codes S0 and S1 which show the initiation, and the data constellation which has a meaning per 1 section as for a sub-code is written in. It is drawing 19 (b) which makes one frame one line and reput in order the data inputted by time series. The data of one symbol are shown by s, and (f, b) among drawing, and s is [a frame number and b of a section number and f] cutting tool numbers at this time.
Drawing 20 is CD and the format Fig. of the error correcting code of CD-ROM which were defined by specification.
C1 and C2 are the correction sections for protecting to a duplex, and they are the error correcting code of Lead Solomon who makes 1 byte both one symbol. C1 consists of code length 32 symbol = data 28 symbol + parity 4 symbols. Moreover, C2 consists of code length 28 symbol = data 24 symbol + parity 4 symbols. Drawing 20 shows the delay relation of the data to which the time amount concerning correction is temporarily supplied as zero, the data inputted into the error correction of C1 and C2, and the data corrected and outputted. The rectangle as which the figure was filled in is a delay circuit which gives frame delay of the numeric value. An inverter expresses all bit flipping.
Even if the time amount concerning correction is disregarded as shown in drawing 20 , when the first data (1, 10, 0) are inputted, data (0, 0, 0) are inputted into C2 correction block for the first time. That is, C2 correction cannot be started, if it is not after the data for at least 108 frames are saved so that clearly from the location of the frame train (0, 10, 0) of drawing 19 (b).
When an error correction was performed according to the specification of CD-ROM, with thin **** showed to drawing 19 (b) (0, 1, 0). (0, 0, 1), ... (0, 1, 2) (0, 1, 30), and (0, 0, 31) are C1 signs of code length 32 symbol, and (0, 3, 1), (0, 8, 2), and .... (1, 9, 27) which were shown with the thick

wire frame (0, 0, 0) are C2 sign of code length 28 symbol.
Next, a format of the error correcting code of DVD defined by specification is briefly explained using drawing 21 . Drawing 21 (a) is drawing showing the time series data after the recovery of a DVD sign, and drawing 21 (b) is the format Fig. of the error correcting code of DVD data.
The signal read from the disk turns into a 8 bits (1 byte) time series signal as shown in drawing 21 (a) through a demodulator circuit. The data equivalent to the sub-code of CD-ROM do not exist, but the signal showing the break of data required for correction is removed in a demodulator circuit. The unit of correction is the matrix of 182 trains of 208 lines, and calls this an ECC (Error Correction Code) block. An ECC block is the error correcting code of Lead Solomon who makes 8 bits (1 byte) one symbol, and is the product code which added parity independently of the line writing direction (inner sign parity) and the direction of a train (outside sign parity). The data of one symbol are shown by (b, r, c) among drawing, and b is [a line number and c of a block number and r] row numbers at this time. Making into one line the data (b, r0) of 182 symbols by which the time series in-and-out force is carried out, (b, r1), and .... (b, r2) (b, r181), each line consists of inner sign parity of the body data of 172 symbols, and ten symbols. Moreover, 208 symbols (b0, c) of the direction of a train, (b1, c), (b2, c), and .... (b207, c) consist of sign parity outside the body data of 192 symbols, and 16 symbols. However, one inserting a line of lines of outside sign parity is made serial to body data of 12 lines. Inner sign correction is performed per line and it performs 208 lines and 182 trains of outside sign corrections per train, and if needed, an error correction performs disappearance correction etc. and is completed.
As stated above, it can be said that CD-ROM and error correction processing of DVD are the combination of the 1st error correction processing which can perform the data inputted by time series, without changing the sequence greatly, and the 2nd correction processing which can be performed for the first time since a certain amount of period data are stored. The 1st correction processing puts C1 correction processing in CD-ROM, puts inner sign correction processing in DVD, and the 2nd correction processing puts C2 correction processing in CD-ROM, and it puts outside sign correction processing in DVD. Such an error correction format does not stop at other recording system media, but is widely applied also to broadcast system media, such as a digital television set.
An error correction cannot be performed, unless the data of a certain amount of [CD-ROM or DVD] amount are stored so that clearly from the specification of the above-mentioned error correcting code. That is, in CD-ROM, if it is going to correct (0, 0, 0), (0, 3, 1), (0, 8, 2), and Cof code length 28 symbol of .... (1, 9, 27) 2 sign, it is necessary to hold the data for 108 frames from (0, 0, 0) to (1, 9, 27). Also in DVD, since the data of 208 symbols (b0, c) of the direction of a train, (b1, c), (b2, c), and .... (b207, c) are needed in order to correct an outside sign, it is necessary to hold the data for 1ECC block at worst.
Conventionally, as a means (data buffer) to hold data, SRAM of 2kByte(s) to 4kByte(s) has been used for CD-ROM, and DRAM of 256kByte(s) to 512kByte(s) has been used for DVD. When it is going to develop CD-ROM and the equipment which can both reproduce DVD, having SRAM of 2kByte(s) further for the error correction of CD-ROM has the technical problem that it is useless since both media are not reproduced to coincidence, and a cost rise is caused, having DRAM of 256kByte(s) to 512kByte(s) required for the error correction of DVD. Moreover, it is also from the economical reason for holding down cost that not SRAM but DRAM is used for the DVD regenerative apparatus.
The rise of cost can be suppressed, if it is able to enable it to operate also considering DRAM of 256kByte(s) to 512kByte(s) required for the error correction of DVD as a data buffer of CD-ROM when developing the regenerative apparatus corresponding to both CD-ROM and DVD. However, although the data access in the same ROW address once DRAM publishes the ROW address is high-speed, access cycles, such as another ROW address issue, are needed for access in different ROW at an excess, and access becomes slow. Access of read-out [especially] for C2 correction of CD-ROM and an output over the ROW address which the address becomes irregular and is different increases, and it has the problem that access takes time amount too much and it is not in time for real-time processing.

Thus, since error correction processing cannot be started unless a certain amount of [in order to be based on the specification of CD-ROM and the error correction of DVD] amount of data is inputted, buffer memory is required and, as for buffer memory, constituting from a DRAM is desirable on constraint of a price. And it enables it to want, as for buffer memory, to be also able to read CD-ROM with DVD drive equipment as common use of DVD and CD-ROM. Furthermore, although DRAM of access in the same ROW address is high-speed, the random access over the different ROW address serves as a low speed extremely.
Then, without solving the technical problem of these former, and being able to use the rapid access in the same ROW address of DRAM, and using expensive memory especially at a high speed, the purpose of this invention is a cheap configuration and is to offer the suitable data processor and suitable art for read-out and the writing for the writing to the buffer memory of the data read from the medium, DVD, the 1st correction processing of CD-ROM, and the 2nd correction processing.
Indication of invention In the data processor of this invention A means to pretreat the signal read from the medium, and the means which writes the pretreated result in DRAM (1st DRAM access), The means which reads the data for the 2nd correction from said DRAM (2nd DRAM access), The means which reads the data for outputting the corrected data to the latter part (3rd DRAM access), It is characterized by having the access-control circuit constituted so that the said 1st, 2nd, and 3rd DRAM accesses might be contained in the same ROW address over several words, respectively.
Moreover, by the data-processing approach of this invention, the data accessed by read-out from the writing from an input circuit to DRAM and DRAM for the 2nd correction processing are characterized by performing the DRAM access control on DRAM which is assigned in the same ROW address, respectively and publishes the ROW address and the column address in this ROW continuously by these numbers of words in each access in the unit of suitable numbers of words.

Since it can respond to the error correction of the signal from a medium which can use high-speed access in the same ROW address of DRAM by this, and is different in common, it is not necessary to use expensive memory at high speed, and parallel processing of the error correction of DVD can be carried out to CD-ROM. Moreover, since the circuits of correction processing including DRAM access can be diverted to the system which changes with some modification by designing the input circuit and output circuit linked to an internal bus so that it may be adapted for a system, development cost is also reducible.
The best gestalt for inventing In order to explain this invention to a detail more, this is explained according to an attached drawing.
Drawing 1 is the block diagram of the data processor in which the 1st example of this invention is shown.
The buffer register 14 for an input circuit 11, an output circuit 12, and data processing is mutually connected with DRAM18 used for the buffer of data through the internal bus 13. The data transfer demand signal from each is inputted into the priority judging circuit 16, and the data according to the judgment result are outputted to an internal bus 13, and address generation which followed the judgment result of priority in the address generation access-control circuit 17 is performed, and the access control of DRAM18 is performed.
To the signal inputted from the medium, pretreatment suitable for a medium is performed, and it accumulates in an input circuit 11 until it becomes the collected numbers of words which are written in DRAM18. The numbers of words of the data continued and written in in the same ROW address by page mode access of a Synchronous DRAM or the usual DRAM are pointed out with "the collected numbers of words" here. When the data of numbers of words required for an input circuit 11 are stored, a write request signal is sent to the priority judging circuit 16. When a demand is received, in the address generation access-control circuit 17, the address generation for input data writing is performed, and the data outputted to the internal bus 13 from the input circuit 11 are written in DRAM.
When the data of a complement are stored in correction at DRAM18, a data demand signal is outputted to a control circuit 17 from the data-processing circuit 15. If the data demand signal

for correction is received by the control circuit 17, the address generation of read-out for correction will be performed, and required data will be stored in a buffer register 14 through an internal bus 13. The data-processing circuit 15 is beginning to read required data serially from a buffer register 14, and performs required processing of an error correction etc.
The location and value of an error which were discovered access DRAM18 per symbol, and are corrected. The corrected data are sent out to the latter part with the protocol defined through the output circuit 12. An output circuit 12 also sends a data demand signal to the priority judging circuit 16 if needed.
The priority judging circuit 16 operates a system, it being efficient and protecting constraint of a real-time operation by judging the priority of the above-mentioned memory access demand, and permitting access sequentially from what has a high priority. For example, the demand from an input circuit 11 is usually processed by top priority. It is because it is not lost before the inputted data are processed. Besides having illustrated, refresh of DRAM18 is also a high demand of a priority.
Drawing 2 is the detail block diagram of the input circuit in drawing 1 .
When it constitutes the equipment which considers the error correction of the signal of DVD as CD-ROM, as shown in drawing 2 , it is constituted as a suitable example of an input circuit 11. The data inputted through a demodulator circuit from CD-ROM are 1 symbol (8 bits) unit, and the signal which shows the head of the clock which synchronized with it, and a frame is inputted. Recovery data are rearranged into the sequence suitable for C1 correction through the 1st data rearrangement circuit 111, and are sent to C1 correction circuit 114 and the 2nd data rearrangement circuit 112. Although all correction processings may be performed in C1 correction circuit 114, only syndrome generation may be performed at worst and the syndrome may be transmitted to a correction arithmetic circuit (illustration abbreviation). At this time, error occurrence frequency is not so high as 1 symbol extent in 1,000 symbols by the usual CD-ROM disk. Since the syndrome of a data stream without an error becomes zero, if only a syndrome with an error will be transmitted to a correction processing circuit, a correction processing circuit can be used more efficiently.
On the other hand, for the error correction of DVD, the signal showing the head of the recovery clock which synchronized with the data to which it restored, and it, and an ECC block is inputted. Furthermore, if the line address signal is inputted, it can use for bailout when a line address becomes irregular within an ECC block by a certain cause. The error correction of DVD can perform correction processing by inputting into the inner sign correction circuit 115 with the sequence of the signal inputted into time series from the demodulator circuit. Therefore, unlike the case of CD-ROM, there is no need for the 1st data rearrangement circuit 111. Moreover, the 2nd data rearrangement circuit 112 may also become unnecessary by devising addressing of DRAM.
Drawing 3 is drawing showing the example of a configuration of the 1st data rearrangement circuit of a CD-ROM recovery signal, and a counter in drawing 2 , and drawing 4 is the timing chart.
The data to which it restored are inputted synchronizing with the recovery clock. 33 ** counter 119 carries out counting of the recovery clock of CD-ROM, and outputs enumerated data i. Enumerated data i are reset whenever a frame head signal is inputted, and they output the value of 0 to 32. At the time of i= 0, since the recovery data of CD-ROM are a sub-code, they are outputted as it is. i= 1, 3 and 5, and ... 11, 17, 19, and ... it was shown in drawing 20 at the time of 27 -- as -- one-frame delay -- being unnecessary -- since it is data, Terminal a is chosen and outputted by the selection circuitry 122. The data which Terminal b was chosen by the selection circuitry 122 and reversed are outputted at the time of i= 13, and 15, 29 and 31. i= 2, 4 and 6, and ... it outputs via the one-frame delay circuit 121 at the time of 32. i= -- 2, 4, and ... 12, 18, 20, and ... since it is required data of one-frame delay at the time of 28 as shown in drawing 20 , Terminal c is chosen and outputted by the selection circuitry 122. As for i= 14, and 16, 30 and 32, the data with which Terminal d was chosen and reversed by the selection circuitry 122 are outputted. the one-frame delay circuit 121 -- i= 2, 4 and 6, and ... it consists of 16 steps of shift registers into which a clock is inputted only at the time of 32. By passing the circuit shown in

drawing 3 , data required for C1 correction are supplied in suitable sequence.
The signal currently outputted to the terminals a, b, c, and d in drawing 3 is shown in the terminals a, b, c, and d of drawing 4 . The output of a selection circuitry 122 is shown as data corrected [C1], and serves as data of sequence suitable for C1 correction. The clock which synchronized with data corrected [C1] is also outputted. Moreover, since a sub-code is used in another network with processing of the usual error correction, it is outputted from another terminal.
It is necessary to devise and to constitute the 2nd data rearrangement circuit 112 so that it can be adapted for the configuration of DRAM used as a buffer. DRAM has the configuration of 8 bits / 1 word, 16 bits / 1 word, and 32 bits / 1 etc. word. Since a correction unit is one symbol, i.e., 8 bits, CD-ROM and DVD become access of 1 in all, 2, and 4 symbol to the word length of DRAM.
When using DRAM (8 bits / 1 word), the 2nd data rearrangement circuit 112 is unnecessary, devises addressing of DRAM as it is, and should just write in data corrected [C1].
Drawing 5 is drawing showing the data allotment approach on 8-bit [/word] DRAM of CD-ROM data.
-1, and (97, 31) are written in the address H'000 in the 1st ROW of DRAM, H'021, and H'042, ..., H'3FF. [-1, (97, 1), (0, 0, 2) and ... which were outputted as data corrected / C1 / (0, 0 0), and] Here, although all bit flipping of cutting tool number =12 of (s:section number, f:frame number, and b:cutting tool number) and the data of 13, 14, 15, 28, 29, 30, and 31 is carried out, since it is not important for data (s, f, b) that it is especially reversal data, they do not distinguish this on a notation. moreover, H'X used with the notation of the address -- X -- hexadecimal table ** -- a certain thing is expressed. The data (0, 1, 0) of the following frame corrected [C1], (0, 0, 1), (0, 1, 2), ..., (0, 0,341) are written in the address H'400 in.the 2nd ROW of DRAM, H'421, and H'442, ..., H'7FF. If it writes in the 3rd and 4th ROW similarly, next, return, data (0, 4, 0) corrected [C1], (0, 3, 1), (0, 4, 2), ..., (0, 3, 31) will be written in the 1st ROW at the address H'3E0 in the 1st ROW of DRAM, H'001, and H'022, .., H' 3DF. Although it is thought that the address becomes discontinuity and a address generation circuit becomes complicated apparently, the column address of DRAM is easily realizable by adding H'021 (it being 33 at a decimal) to a front value one by one using the 10-bit adder which disregarded overflow.
If data are written in by the approach explained above, a series of data required for C2 correction will be arranged by the address H'000 which continues as shown in drawing 5 , H'001, H'002, ..., H'01F. 81F, address H'C00, H'C01, H'C02, ..., H'C1F are followed. the following -- one by one -- the address H'400, H'401, H'402, ..., H -- '41F, the address H'800, H'801, H'802, ..., H' -- A series of data for C2 correction are written in the address H'3E0, H'3E1, H'3E2, ..., H'3FF. Since it is consecutive addresses, the configuration of an address-generation circuit is easy. Although explained in this example writing the cutting tool numbers 28, 29, 30, and 31 in DRAM, since these are the parity for C1 correction, as long as there are no reasons of performing C1 correction again after C2 correction, there is especially no need for writing.
Read-out of the data which correction ended needs to give DIN TARIBU defined by specification as shown in drawing 20 .
DIN TARIBU is realizable by addressing of DRAM read-out with the data assignment on DRAM of this example.
2-byte delay is everywhere included in DIN TARIBU of drawing 20 .
This is equivalent to the difference of the 1st, the 3rd ROW, or the 2nd and the 4th ROW on DRAM. Therefore, (0, 3, 1) in drawing 20 (0, 0, 0), (0, 24, 6), (0, 27, 7), (0, 62, 16), (0, 65, 17), (0, 86, 22), (0, 89, 23), (0, 8, 2), (0, 11, 3), (0, 32, 8), (0, 35, 9), (0, 70, 18), (0, 73, 19), (0, 94, 24), (0, 97, 25), (0, 16, 4), (0, 19, 5), (0, 40, 10), When outputting (0, 43, 11), (0, 78, 20), (0, 81, 21), (1, 4, 26), and (1, 7, 27), it is attained by repeating renewal of the ROW address, and read-out in [of 4 bytes] the ROW address 6 times as follows.
1: The 1st ROW The address H'000, H'001, H'006, H'0072: The 3rd ROW The address H'810, H'811, H'816, H'8173: The 1st ROW The address H'002, H'003, H'008, H'0094 : [The 3rd ROW] The address H'812, H'813, H'818, H'8195: The 1st ROW The address H'004, H'005, H'00A, H'00 B6: The 3rd ROW The address H'814, H'815, H'81A, and H'81B Since the above-mentioned

access accesses the 1st ROW and the 3rd ROW by turns Although it is hard to say that it is not necessarily efficient, effectiveness is in a simple configuration and cost reduction in that did not have special hardware but processing (refer to drawing 20) of DIINNTARIBU required for an output is realized only by addressing of DRAM.
Drawing 13 is the block diagram showing the 1st example (for 8-bit [/word] DRAM) of the output circuit in drawing 1 .
Addressing is divided into 2 sets, the 1st ROW and the 3rd ROW, an output circuit as shown in drawing 13 can be added, and DIN TARIBU can also be realized.
That is, as shown below, it reads from the 1st ROW shown in drawing 5 , and the 3rd ROW every 12 words, respectively.
1: The 1st ROW Address H'H[from 000]'00 B-2: The 3rd ROW From the address H'810 to H'81B The read data are transmitted to the output circuit shown in drawing 13 through the internal bus. A data transfer is performed synchronizing with a read-out clock, and the address of the register which should be written in with 24 ** counter 131 of the input side in an output circuit is specified. When the first data are inputted, 24 ** counter 131 is 0, and whenever a read-out clock is inputted, it is added every [1]. The output of a decoder 132 publishes write enable so that data may be stored in those with 24 from 0 to 23, and 24 output latches 134 in order of an output corresponding to a counter value. After all data are stored in the output latch 134, whenever an output request clock is inputted, the data of one symbol are outputted.
In drawing 13 , although the connection place of the output of the decoder 136 of an output side is not specified, it is designed so that it may be outputted sequentially from a top corresponding to the counted value of 24 ** counter 137 of an output side. In this example, an input side is devised, and it has designed so that data may be stored in a latch circuit 134 in order of an output. Therefore, since an output is a sequential access, it may adopt the format of a shift register. Moreover, it is also possible to store the input side in the order read from DRAM, and to make it the design outputted in suitable sequence.
Since addressing of three sorts of DRAM accesses of read-out for the input data writing of CD-ROM described above and C2 correction and the output of corrected data tends
[comparatively] to find out regularity, it can constitute from an easy register and an easy adder.

Drawing 6 is drawing showing the example of the data allotment on the bit / WORD DRAM of the DVD data in this invention.
The approach to assign the data in the case of using the same DRAM for the error correction of the signal read from DVD is shown by drawing 6 . The data (0, 0, 0) of the ECC block of the 1st line serially inputted from a demodulator circuit, (0, 0, 1), and ... (0 0,181) are written in the 6th ROW one by one from the 1st ROW of DRAM. (0, 0, 31) are stored in the 1st ROW from data (0, 0, 0) by H'001F from the address H'0000. (0, 0, 63) are stored by H'041F from the address H'0400 of the 2nd ROW from the following data (0, 0, 32), and data are similarly written in one by one from the head of the 3rd, 4th, 5th, and 6th ROW.
Since the data for the ECC block of one line cannot be divided among 182 bytes 32, H'141F become a free space from the address H'1416 of the 6th ROW. The data (0, 1, 0) of the ECC block of the 2nd line which continue and are serially inputted from a demodulator circuit, (0, 1, 1), and ... (0 1,181) return to the 1st ROW of DRAM, and it is written in by H'003F from the address H'0020 of a continuation of data of the 1st line. Data of the 32nd line are similarly written in the 6th ROW one by one from the 1st ROW of DRAM from the ECC block of the 3rd line. Data of the 64th line from the ECC block of the 33rd line furthermore, to the 12th ROW from the 7th ROW of DRAM Data of the 96th line from the 65th line to the 18th ROW from the 13th ROW of DRAM Data of the 128th line from the 97th line to the 24th ROW from the 19th ROW of DRAM Data of the 160th line from the 129th line to the 30th ROW from the 25th ROW of DRAM Data of the 192nd line from the 161st line to the 36th ROW from the 31st ROW of DRAM As the remaining data of the 193rd line to the 208th line wrote data of the 32nd line in the 6th ROW from the 1st ROW of DRAM from the ECC block of the 1st line from the 37th ROW of DRAM at the 42nd ROW, respectively, it writes in one by one.
However, the field of data of the 228th line is virtually assigned to the 42nd ROW from the last

ROW [37th] from the ECC block of the 209th line which originally does not exist from the semantics which maintains the regularity of the address. In fact, since such data are not inputted, this field also becomes intact.
Although the example shown in drawing 2 showed the example which performs activation of all or a part of inner sign corrections before storing data in DRAM According to allotment of the data shown in drawing 6 , 32 symbols of a series of 182 symbols required for inner sign correction are stored at a time in the same ROW address of DRAM. Since it can read without the recurrence line of the ROW address continuously up to a maximum of 32 symbol, after storing data in DRAM, inner sign correction may be performed.
A series of data required for outside sign correction are (b0, c), (b1, c), and ... (b207, c). As shown in drawing 6 (0, 0, 0), (0, 1, 0), and ... (0, 31, 0) are stored in the 1st ROW of DRAM, and it can read without the recurrence line of the ROW address continuously up to a maximum of 32 symbol. therefore, a series of data (0, 0, 0) required for outside sign correction, (0, 1, 0), and ... (0, 207, 0) are read -- being alike -- what is necessary is just to perform the following steps 1: The 1st ROW H'0000, 0020 and 0040, ..., 03E02: The 7th ROW H'1800, 1820 and 1840, ..., 1BE03: The 13th ROW H'2000, 2020 and 2040, ..., 23E04 : [The 19th ROW] H'2800, 2820 and 2840, ..., 2B E05: The 25th ROW H'3000, 3020 and 3040, ..., 33E06: The 31st ROW H'3800, 3820 and 3840, ..., 3BE07 : [The 37th ROW] H'4000, 4020 and 4040, ..., 41E0 Although 32-byte continuation is possible for it after the step of 1-6 publishes the ROW address After the last step [7th] publishes the ROW address, read-out of 16 bytes or more becomes access of a free space and is not desirable.
The same procedure can also perform read-out of 2 train henceforth of an ECC block.
What is necessary is just to perform read-out of the data after correction in the completely same sequence as the first writing.
Although the data allotment only for 1ECC block was shown in drawing 6 , it is common to it to store the data of 3ECC block on DRAM, and to process them at worst, with usual equipment. That is, whenever it classifies the 2nd ECC block into access for correction, it classifies the 3rd ECC block into read-out for an output, respectively and processing completes the 1st, 2nd, and 3rd ECC block in the writing of data, the 1st ECC block is used for it in pipeline, and makes **** it. In order to take the allowances of error correction time amount, it is rubbed when storing and processing the data beyond 4ECC block sequentially, and there is nothing then.
Drawing 7 is drawing showing one example of the 2nd data rearrangement circuit in the case of using DRAM (16 bits / 1 word).
As for even (i= 1, 3, 5, ... 31) data, a cutting tool number gives four-frame delay, and a cutting tool number outputs odd (i= 2, 4, 6, ... 32) data without delay to a high-order byte side to a lower byte side. As shown in the timing chart of drawing 4 , data (0, 0, 0) are combined with data (0, 3, 1), and are outputted. 2 bytes (16 bits) outputted by receiving delay and combining in the 2nd data rearrangement circuit are written in 1 word of DRAM.
Drawing 8 is drawing showing an approach to assign the data on 16-bit [/word] DRAM of CD-ROM data.
data (0, 0, 0) and 2 bytes of data made in the combination of (0, 3, 1) -- {(0, 0, 0), if it is written as} (0, 3, 1) 16 words outputted by time series -- {(0, 0, 0) --} (0, 3, 1) -- {(0, 0, 2) -- (0, 3, 3)} -- {(0, 0, 4) --} (0, 3, 5) and ... {(0, 0, 30) --} (0, 3, 31) -- the address H in the 1st ROW of DRAM -- '000, H' -- 011, H'022, and ... H' -- it writes in 0FF. then, data -- {(0, 1, 0) --} (0, 4, 1) -- {(0, 1, 2) -- (0, 4, 3)} -- {(0, 1, 4) --} (0, 4, 5) and ... {(0, 1, 30) --} (0, 4, 31) -- the address H in the 2nd ROW of DRAM -- '100, H' -- 111, H'122, and ... H' -- it writes in 1FF. data after writing this in to the 8th ROW of DRAM one by one -- {(0, 8, 0) -- (0, 11, 1)} -- {(0, 8, 2) --} (0, 11, 3) -- {(0, 8, 4) -- (0, 11, 5)} and ... {(0, 8, 30) --} (0, 11, 31) -- again -- the address H in the 1st ROW of DRAM -- '0F0, H' -- 001, H'012, and ... H' -- it writes in 1EF. That the ROW address should just repeat from the 1st to the 8th, the COLUMN address gives initial value suitable whenever the ROW address changes, after that, disregards overflow and should just add it H'11 (it is 17 at a decimal) every. Consequently, they are a series of 32 symbols required for C2 correction 16 words which continues from H'0 4 bits of low order of the address so that clearly also from drawing 8 .
Read-out of the data which correction ended needs to give DIN TARIBU defined by specification

as shown in drawing 20 . DIN TARIBU is realizable by addressing of DRAM read-out with the data assignment on DRAM of this example. 2-byte delay is everywhere included in DIN TARIBU of drawing 20 .
This is equivalent to the 1st, the 3rd and the 2nd, the 4th and the 3rd, the 5th and the 4th, the 6th and the 5th, the 7th and the 6th, the 8th and the 7th, the 1st or the 8th, and the difference of the 2nd ROW on DRAM. Therefore, (0, 3, 1), (0, 24, 6) in drawing 20 (0, 0, 0), (0, 27, 7), (0, 62, 16), (0, 65, 17), (0, 86, 22), (0, 89, 23), (0, 8, 2), (0, 11, 3), (0, 32, 8), (0, 35, 9), (0, 70, 18), (0, 73, 19), (0, 94, 24), (0, 97, 25), (0, 16, 4), (0, 19, 5), (0, 40, 10), When outputting (0, 43, 11), (0, 78, 20), (0, 81, 21), (1, 4, 26), and (1, 7, 27), it is attained by repeating renewal of the ROW address, and read-out in [of 4 bytes] the ROW address 6 times as follows.
1: The 1st ROW The address H'000, H'0032: The 3rd ROW The address H'108, H'10B3: The 1st ROW The address H'001, H'0044: The 3rd ROW Address H'109, H'10C5 : [The 1st ROW] The address H'002, H'0056: The 3rd ROW Address H'10A, H'10D The above-mentioned access Since the 1st ROW and the 3rd ROW are accessed by turns, although it is hard to say that it is not necessarily efficient, the point of having not had special hardware but having realized processing (referring to drawing 20) of DIINNTARIBU required for an output only by addressing of DRAM is effective.
On the other hand, addressing is divided into 2 sets, the 1st ROW and the 3rd ROW, an output circuit as shown in drawing 14 can be added, and DIN TARIBU can also be realized. As shown below, it reads every 6 words the 3rd ROW with the 1st ROW, respectively.
1: The 1st ROW From the address H'000 to H'0052: The 3rd ROW Address H'Hfrom 108'10D The read data are transmitted to the output circuit shown in drawing 14 through the internal bus. A data transfer is performed synchronizing with a read-out clock, and the address of the register which should be written in with the duodecimal counter 141 of the input side in an output circuit is specified. When the first data are inputted, the duodecimal counter 141 is 0, and whenever a read-out clock is inputted, it is added every [1]. The output of a decoder 142 publishes write enable so that data may be stored in those with 12 from 0 to 11, and 24 output latches 144 in order of an output corresponding to the value of a counter 141. After all data are stored in latch 144, whenever an output request clock is inputted, the data of one symbol are outputted. In drawing 14 , although the connection place of the output of the decoder 146 of an output side is not specified, it is designed so that it may be outputted sequentially from a top corresponding to the counted value of 24 ** counter 147 of an output side. In this example, an input side is devised, and it has designed so that data may be stored in a latch circuit 144 in order of an output. Therefore, since an output is a sequential access, it may adopt the format of a shift register. Moreover, the input side is stored in the order read from DRAM, and the design outputted in suitable sequence is also possible.
Since addressing of three sorts of DRAM accesses of read-out for the input data writing of CD-ROM described above and C2 correction and the output of corrected data tends [comparatively] to find out regularity, it can constitute from an easy register and an easy adder.

Drawing 9 is drawing showing an approach to assign the data in the case of using the 16 bits [/word] same DRAM for the error correction of the signal read from DVD.
The data (0, 0, 0) of the ECC block of the 1st line serially inputted from a demodulator circuit, (0, 0, 1), and ... (0 0,181) are written in the 6th ROW one by one from the 1st ROW of DRAM. However, two continuous symbols are collectively used as 16 bits = 1 word data. the 1st ROW -- from the address H'0000 by H'000F -- data -- {(0, 0, 0) -- from} (0, 0, 1) -- {(0, 0, 30) --} (0, 0, 31) is stored. the following data -- {(0, 0, 32) -- from} (0, 0, 33) -- {(0, 0, 62) --} (0, 0, 63) -- address H' of the 2nd ROW -- it is stored by H'010F from 0100, and data are similarly written in one by one from the head of the 3rd, 4th, 5th, and 6th ROW.
Since the data for the ECC block of one line cannot divide 182 bytes among 32, H'050F become a free space from address H'050B of the 6th ROW. the data of the ECC block of the 2nd line which continue and are serially inputted from a demodulator circuit -- {(0, 1, 0) --} (0, 1, 1) and ... {(0 1,180) --} (0 1,181) It returns to the 1st ROW of DRAM and is written in 2nd ROW address H 'sequential 6th ROW address H from 0110' 051A from the address H'0010 of a continuation of

data of the 1st line to H'001F. Similarly, data of the 16th line are written in the 6th ROW one by one from the 1st ROW of DRAM from the ECC block of the 3rd line. Data of the 32nd line from the ECC block of the 17th line furthermore, to the 12th ROW from the 7th ROW of DRAM Data of the 48th line from the 33rd line to the 18th ROW from the 13th ROW of DRAM As the last data of the 193rd line to the 208th line wrote data of the 32nd line in the 6th ROW from the 1st ROW of DRAM from the ECC block of the 1st line from the 73rd ROW of DRAM at the 78th ROW, respectively, it writes in one by one.

Although the example shown in drawing 2 showed the example which performs all or a part of inner sign corrections before storing data in DRAM According to allotment of the data shown in drawing 9 , 32 symbols of a series of 182 symbols required for inner sign correction are stored at a time in the same ROW address of DRAM. Since it can read without the recurrence line of the ROW address continuously up to a maximum of 32 symbol, after storing data in DRAM, inner sign correction may be performed.

A series of data required for outside sign correction are (b0, c), (b1, c), and ... (b207, c). As shown in drawing 9 (0, 0, 0), (0, 1, 0), and ... (0, 15, 0) are stored in the 1st ROW of DRAM, and it can read without the recurrence line of the ROW address continuously up to a maximum of 16 symbol. Since it is 2 bytes/word and it can also read (0, 1, 1), and ... (0, 15, 1) to coincidence at this time (0, 0, 1), DRAM is efficient if sign correction is processed to juxtaposition outside the 0th train and the 1st train. The generating approach of the address of DRAM can be easily constituted like the case of 8 bits [/word] DRAM.

What is necessary is just to perform read-out of the data after correction in the completely same sequence as the first store.

Although the data allotment only for 1ECC block was shown in drawing 9 as well as drawing 6 , it is common to store the data of 3ECC block on DRAM, and to process them at worst, with usual equipment, as above-mentioned.

Drawing 10 is the block diagram of the 2nd data rearrangement circuit showing the 2nd example of this invention, and shows the case where DRAM (32 bits / 1 word) is used.

As for the data of the multiple (i= 1, 5, 9, ... 29) of 4, a cutting tool number gives 12-frame delay. A most significant byte side, A number (i= 2, 6, 10, ... 30) of data with which a cutting tool number divides by 4, and remains one give 12-frame delay. To the 2nd high-order byte A number (i= 3, 7, 11, ... 31) of data with which a cutting tool number divides by 4, and remains two give eight-frame delay, and a number (i= 4, 8, 12, ... 32) of data which a cutting tool number divides by 4 to the 3rd high-order byte, and remain three are made into a total of 4 bytes to the lowest cutting tool, and they output them without delay to him. As shown in the timing chart of drawing 4 , data (0, 0, 0) (0, 3, 1) (0, 8, 2) (0, 11, 3) are put together and outputted. Delay is received, and it combines, and is outputted in the 2nd data rearrangement circuit, and 4 bytes (32 bits) are written in 1 word of DRAM.

Drawing 11 is drawing showing an approach to assign the data on 32-bit [/word] DRAM of CD-ROM data.

4 bytes of data made in the combination of data (0, 0, 0), (0, 3, 1), (0, 8, 2), and (0, 11, 3) -- {(0, 0, 0), if it is written as (0, 3, 1), (0, 8, 2), and} (0, 11, 3) 16 words outputted by time series -- {(0, 0, 0) -- (0, 3, 1) -- (0, 8, 2) (0, 11, 3), and} -- {(0, 0, 4) -- (0, 3, 5) -- (0, 8, 6) (0, 11, 7),}, and ... {(0, 0, 28) -- (0, 3, 29), (0, 8, 30), and} (0, 11, 31) -- the address H in the 1st ROW of DRAM -- '000, H' -- 009, H'012, and ... it writes in H'03F. then, data -- {(0, 1, 0) -- (0, 4, 1), and (0, 9, 2) -- (0, 12, 3)} -- {(0, 1, 4) --} (0, 4, 5), and (0, 9, 6) -- (0, 12, 7)} and ... {(0, 1, 28) -- (0, 4, 29), (0, 9, 30), and} (0, 12, 31) -- address H' in the 2nd ROW of DRAM -- 040, H'049, and ... it writes in H'07F. data after writing this in to the 16th ROW of DRAM one by one -- {(0, 16, 0) -- (0, 19, 1), (0, 24, 2) (0, 27, 3),}, and ... {(0, 16, 28) -- (0, 19, 29), (0, 24, 30) (0, 27, 31), and} -- again -- the address H in the 1st ROW of DRAM -- '038, H' -- 001, H'012, and ... it writes in H'036. That the ROW address should just repeat from the 1st to the 16th, the COLUMN address gives initial value suitable whenever the ROW address changes, after that, disregards overflow and should just add it H'11 (it is 17 at a decimal) every. Consequently, they are a series of 32 symbols required for C2 correction 4 words which continues from the low order triplet H'0 of the address so that clearly also from drawing 11 .

Read-out of the data which correction ended needs to give DIN TARIBU defined by specification as shown in drawing 20 . DIN TARIBU is realizable by addressing of DRAM read-out with the data allocation on DRAM of this example. 2-byte delay is everywhere included in DIN TARIBU of drawing 20 . this -- a DRAM top -- the 1st, the 3rd and the 2nd, the 4th and the 3rd, the 5th and the 4th, the 6th and the 5th, the 7th and the 6th, and the 8th ... it is equivalent to the 15th, the 1st or the 16th, and the difference of the 2nd ROW. Therefore, (0, 3, 1), (0, 24, 6) in drawing 20 (0, 0, 0), (0, 27, 7), (0, 62, 16), (0, 65, 17), (0, 86, 22), (0, 89, 23), (0, 8, 2), (0, 11, 3), (0, 32, 8), (0, 35, 9), (0, 70, 18), (0, 73, 19), (0, 94, 24), (0, 97, 25), (0, 16, 4), (0, 19, 5), (0, 40, 10), When outputting (0, 43, 11), (0, 78, 20), (0, 81, 21), (1, 4, 26), and (1, 7, 27), it is attained by repeating renewal of the ROW address, and read-out in [of 3 words] the ROW address twice as follows. 1: 1st ROW The address H'000, H'001, H'0022: The 3rd ROW address H'084, H'085, H'086 Since it is 32 bits/word, while there are few counts of access and they end, they cannot perform exchange of the data (symbol) of a cutting tool unit only in addressing.
Drawing 15 is the block diagram (for 32-bit [/word] DRAM) of an output circuit showing the 3rd example of this invention.
In order to realize output order as specification, the DIN TARIBU circuit as shown in drawing 15 is needed. The read data are transmitted to the output circuit shown in drawing 15 through the internal bus. A data transfer is performed synchronizing with a read-out clock, and the address of the register which should be written in with 6 ** counter 152 of the input side in an output circuit is specified. Since it is four symbols per word, the latch 153 written in is specified as four-piece coincidence. When the first data are inputted, 6 ** counter 152 is 0, and whenever a read-out clock is inputted, it is added every [1]. The output of a decoder 151 publishes write enable so that data may be stored in those with six from 0 to 5, and 24 output latches 153 in order of an output corresponding to the value of a counter 152. After all data are stored in latch 153, whenever an output request clock is inputted, the data of one symbol are outputted.
Although the connection place of the output of the decoder 155 of an output side is not specified in drawing 15 , it is designed so that it may be outputted sequentially from a top corresponding to the counted value of 24 ** counter 156 of an output side. In this example, an input side is devised, and it has designed so that data may be stored in a latch circuit in order of an output. Therefore, since an output is a sequential access, it may adopt the format of a shift register.
Since addressing of three sorts of DRAM accesses of read-out for the input data writing of CD-ROM described above and C2 correction and the output of corrected data tends [comparatively] to find out regularity, it can constitute from an easy register and an easy adder.

Drawing 12 is drawing showing an approach to assign the data in the case of using the 32 bits [/word] same DRAM for the error correction of the signal read from DVD.
The data (0, 0, 0) of the ECC block of the 1st line serially inputted from a demodulator circuit, (0, 0, 1), and ... (0 0,181) are written in the 6th ROW one by one from the 1st ROW of DRAM. However, four continuous symbols are collectively used as 32 bits = 1 word data.
the 1st ROW -- from the address H'0000 by H'0007 -- data -- {(0, 0, 0) -- from (0, 0, 1), (0, 0, 2), and} (0, 0, 3) -- {(0, 0, 28) -- (0, 0, 29), (0, 0, 30), and} (0, 0, 31) are stored. the following data -- {(0, 0, 32) -- (0, 0, 33), and (0, 0, 34) -- (0, 0, 35) from} -- {(0, 0, 60) -- (0, 0, 61), (0, 0, 62), and} (0, 0, 63) -- address H' of the 2nd ROW -- it is stored from 0040 by H'0047, and data are similarly written in one by one from the head of the 3rd, 4th, 5th, and 6th ROW.
Since the data for the ECC block of one line cannot divide 182 bytes among 32, H'014F become a free space from 2 bytes of low order of the address H'0145 of the 6th ROW. the data of the ECC block of the 2nd line which continue and are serially inputted from a demodulator circuit -- {(0, 1, 0) -- (0, 1, 1), (0, 1, 2) (0, 1, 3),}, and ... {(0 1,178) -- (0, 1,179), (0, 1,180), and} (0 1,181) It returns to the 1st ROW of DRAM and is written in order order 6ROW address H'004F from the 2nd ROW address H'0048 from the address H'0008 of a continuation of data of the 1st line to H'000F. Data of the 16th line are similarly written in the 6th ROW one by one from the 1st ROW of DRAM from the ECC block of the 3rd line. Data of the 32nd line from the ECC block of the 17th line furthermore, to the 12th ROW from the 7th ROW of DRAM Data of the 48th line from

the 33rd line to the 18th ROW from the 13th ROW of DRAM As the last data of the 193rd line to the 208th line wrote data of the 32nd line in the 6th ROW from the 1st ROW of DRAM from the ECC block of the 1st line from the 151st ROW of DRAM at the 156th ROW, respectively, it writes in one by one.
Although the example shown in drawing 2 showed the example which performs all or a part of inner sign corrections before storing data in DRAM According to allotment of the data shown in drawing 12 , 32 symbols of a series of 182 symbols required for inner sign correction are stored at a time in the same ROW address of DRAM. Since it can read without the recurrence line of the ROW address continuously up to a maximum of 32 symbol, after storing data in DRAM, inner sign correction may be performed.
A series of data required for outside sign correction are (b0, c), (b1, c), and ... (b207, c). As shown in drawing 12 (0, 0, 0), (0, 1, 0), and ... (0, 7, 0) are stored in the 1st ROW of DRAM, and it can read without the recurrence line of the ROW address continuously up to a maximum of 8 symbol.
Since DRAM is 4 bytes/word, at this time The 1st train (0, 0, 1), Since (0, 1, 1), ... (0, 7, 1), the 2nd train (0, 0, 2), (0, 1, 2), ... (0, 7, 2), the 3rd train (0, 0, 3), (0, 1, 3), and ... (0, 7, 3) can also be read to coincidence It is efficient if sign correction is processed to juxtaposition outside the 0th train to the 3rd train. The generating approach of the address of DRAM can be easily constituted like the case of 8 bits [/word] DRAM. What is necessary is just to perform read-out of the data after correction in the completely same sequence as the first store.
Although the data allotment only for 1ECC block was shown in drawing 12 as well as drawing 6 , it is common to store the data of 3ECC block on DRAM, and to process them at worst, with usual equipment, as above-mentioned.
Drawing 16 is the block diagram of the address generation circuit applied to each example of this invention.
It is constituted from the ROW address register 163, COL address register 163a, and a register 0 by Register n (164) as the 1st of an adder circuit 165 and an adder circuit 165 and the 2nd input bus, an output bus, and two or more registers. The value of the ROW address register 163 and COL address register 163a lets a mask / offset circuit 167 pass, respectively, is chosen by the multiplexer 168, and is inputted into the address terminal of DRAM. A mask / offset circuit 166 is constituted using general-purpose ALU (Arithmetic Logic Unit) and a general-purpose register. Since ALU performs the logical operation for every bit, and numeric-values (OR, AND, exclusive OR, etc.) addition and subtraction, the mask which fixes a part of address value to 0 or 1, and the offset adding a constant value are easily realizable. The 1st input bus of an adder circuit 165 chooses one in the ROW address register 163, COL address register 163a, and a register 0 to the register n (164), and inputs it into one side of an adder circuit 165, and the 2nd input bus chooses one in a register 0 to the register n (164), and inputs it into another side of an adder circuit 165.
An addition result is written in one in two or more registers 164,163 through an output bus. Moreover, a direct value is also storable in a register 164,163 through an output bus. Although not shown in drawing, an address-generation circuit can give an operation procedure by the sequence control circuit which received the result of a priority judging circuit.
As stated above, access to DRAM can collect into the continuous word access to the same ROW address by having a suitable input circuit, the 2nd data rearrangement circuit in CD-ROM and a suitable output circuit, especially the DIN TARIBU circuit in CD-ROM according to the number of bits of the WORD of DRAM. For this reason, the high transfer rate using page mode access can be obtained, a working speed is low, namely, DRAM of a low price can also be used effectively. While much number of bits per word of DRAM can reduce the number of access cycles so that there is, a fault, like the scale of the input output circuit whose number of pins increases becomes large generates it. This point is traded off.
Although it is also the same as that of the DRAM external case which it has described above almost when it constitutes DRAM on the same LSI, considering as 8 bits per word is most suitable. In the case of DRAM on chip, there is no need for an address multiplexer, namely, a row address and a column address can be published to coincidence. Moreover, since the rate which

reads the data in the same row address is early enough, changing a column address since the data of the same row address are once put in block and it is read, the 2nd data rearrangement circuit which summarizes two or more symbols is unnecessary in many cases.
Drawing 17 is drawing showing an example of CD-ROM which carried out this invention, and the optical disk regenerative apparatus which can reproduce DVD.
It is constituted combining equipments of an optical-system mechanical system, such as the actuator 171 to which the stage in which the actuator 171 which changes the spindle motor 174 made to rotate a disk, the laser diode 172 which irradiates light at a disk, and the laser diode 172 of 173 or 2 photodetectors which detect the reflected light, the laser diode 172, and the photodetector 173 were installed is moved, and the equipment of an electric system. In CD-ROM and DVD, it will have another laser diode 172 with which wavelength differs, respectively.
As for an electric system, the microcontroller 174 is controlling the whole. The laser control circuit 178 operates, the output of a laser diode 172 is controlled by directions of a microcontroller 174 through the laser driver 176, and the servo signal processing circuit 179 controls a mechanical system by them through the actuator driver 175 and Motor Driver 180.
The signal detected by the photodetector 173 is inputted into the servo processing circuit 179 and the sampling binary-ized circuit 182 through the analog signal processing circuit 177. The signal inputted into the servo signal processing circuit 179 is the error information of tracking or a focus, and is used as a source signal of the location of a stage, fine tuning of a lens, and adjustment of the rotational frequency of a spindle motor. The information made sampling binary is disk information to reproduce.
The clock which synchronized with data from the signal of a photodetector 173 is reproduced, sampling binary-ization is performed using the clock, and a digital signal is acquired. The alignment pattern set to specification from the acquired digital signal is detected, and the initiation timing of the frame of CD-ROM or DVD is identified. A microcontroller 184 supervises the synchronous condition of a playback clock, the condition of alignment pattern detection, and a synchronous condition, and it uses for control of an actuator 171, a servo, and laser and the mode setting of CD-ROM/DVD. The serial digital signal made binary is changed into a parallel signal (17 bits or 16 bits) according to the mode setting of CD-ROM/DVD, respectively, and 8 ****, it gets over 16/8 and it is inputted into 14 / input circuit shown in the above-mentioned example. Since, as for the case of CD-ROM, the triplet is added for 14-bit connection of an EFM (Eight to Fourteen Modulation) modulation, the output of the S/P conversion circuit 185,186 is 17 bits.
Actuation of an input circuit, an output circuit, an address-generation circuit, a priority judging circuit, an error correction processing circuit, DRAM, etc. is as the already shown example. The data [finishing / correction] sent out through an output circuit are transmitted to 188, such as a personal computer, and an image / voice playback device, by the protocol control circuit 187 with the defined protocol.
A microcontroller 184 is assumed to be loaded with a disk with CD-ROM or DVD, chooses a laser diode 172, and operates servo processing, the error correction processing circuit 192, etc. When it differs from the disk loaded with the assumption, it judges that the alignment pattern with which a clock is not reproduced is undetectable, that an error cannot be corrected, etc. from the information which the microcontroller 184 is supervising, and changes to the playback mode of the disk of another side. Thus, automatic distinction of the medium is carried out.
In order to improve correction processing and capacity, the data-processing circuit 15 shown in drawing 1 may consist of parallel processors. It is efficient when the processor element which constitutes a parallel processor is made into n pieces, and C1 sign of CD-ROM of an n-tuple, C2 sign, the inner sign of DVD, and the receiving sign of an outside sign are inputted into juxtaposition. That is, in the case of the inner sign of DVD, in the case of n lines and an outside sign, the data of n train are inputted into n processor elements, respectively, and parallel processing is performed.
Drawing 18 is the block diagram of a data-processing circuit showing one example of this invention, and shows a buffer register in case the number of the processor elements PE is eight, and the configuration of a data-processing circuit.

As for DRAM, 16 bits [word] /, therefore an internal bus also assume 16-bit width of face. Moreover, in playback of CD-ROM, the data allotment on DRAM is as having been shown in drawing 8 , and, in playback of DVD, is as having been shown in drawing 9 .
In drawing 18 , the buffer register consists of 1-byte registers of 1 input 2 output of eight-line eight trains. The data input terminal DB [15:0] is connected to the above-mentioned internal bus, and data are stored in two 1-byte registers of the line chosen from the write enable signal WAR0 of a line by WAR7, and the train chosen from the write enable signal WAC0 of a train by WAC3. On the other hand, read-out is outputted to juxtaposition from eight registers located in a line in the line writing direction or the direction of a train. The direction of a line or a train is specified by control terminal R/C, in the case of [every 1 byte of] a line writing direction, one line and 8 bytes of data which were chosen from RR0 among RR(s)7 are read, and PE7 is supplied from the processor element PE 0 of a data-processing circuit. In the case of [every 1 byte of] the direction of a train, one train and 8 bytes of data which were chosen among RC0 to RC7 are read, and PE7 is supplied from the processor element PE 0 of a data-processing circuit.
The data transfer procedure in the case of performing C2 correction of CD-ROM by the parallel processors PE0-PE7 of drawing 18 is explained. In drawing 8 , 28 bytes required for C2 correction are stored in the address with which it continues from a lower address H'0 to H'D as above-mentioned. 8 bytes of data of H'003 are transmitted to the 1st line of a buffer register from the address H'000 of the 1st ROW of DRAM. continuing -- 8 bytes of data of the address H'100 of the 2nd ROW to H'103 -- the 2nd line of a buffer register -- ** -- 8 bytes of data of H'703 are repeatedly transmitted to the 8th line of a buffer register from the address H'700 of the 8th ROW. A buffer register will be in a full condition now.
The C, i.e., train, side is chosen with a R/C terminal, and every 8 bytes of data are transmitted to PE7 from the processor element PE 0 from RC0 to RC7 one by one. By this actuation, the sequential transfer of the 8th line data of a buffer register is carried out from the processor element PE 0 from the 1st line at PE7, respectively. Since a buffer register becomes empty now, 4 words next to [of DRAM] the 1st ROW to the 8th ROW and a total of 64 bytes are again transmitted to PE7 from the processor element PE 0 through a buffer register. By repeating this, all required data are transmitted to PE7 from the processor element PE 0. In a data-processing circuit, correction processing of juxtaposition of C2 correction with eight processor elements can be performed. Data required for C2 correction are 28 bytes, and since a transfer unit is 8 bytes, in the 4th transfer, 4 bytes becomes data with the unnecessary last. Since these 4 bytes are C1 parity, there is no need in correction of C2. Then, it can also use as a status flag of C1 correction using these 4 bytes.
Next, the data transfer procedure in the case of performing inner sign correction of DVD by the parallel processors PE0-PE7 of drawing 18 is explained.
In drawing 9 , 182 bytes required for inner sign correction are stored in the address with which it continues from the lower address H'0 of the 1st ROW to the 6th ROW of DRAM to H'F every 32 bytes as above-mentioned. From the address H'0000 of the 1st ROW of DRAM, 8 bytes of data of H'0013 are repeated from the address H'0010 to the 1st line of a buffer register, this is repeated for 8 bytes of data of H'0003 to the 2nd line of a buffer register, and 8 bytes of data of H'0073 are transmitted to the 8th line of a buffer register from the address H'0070. Thereby, 8 bytes of the beginning from the 1st line of an ECC block to the 8th line were transmitted by the 8th line from the 1st line of a buffer register. The C, i.e., train, side is chosen with a R/C terminal, and every 8 bytes of data are transmitted to PE7 from the processor element PE 0 from RC0 to RC7 one by one. The sequential transfer of the data from the 1st line of a buffer register to the 8th line is carried out from the processor element PE 0 by this actuation at PE7, respectively. since a buffer register becomes empty now -- again -- H'0007 from the next address H'0004 of the 1st ROW of DRAM, H'0014 to H'0017, and ... a total of 64 bytes of H'Hfrom 0074'0077** are transmitted to PE7 from the processor element PE 0 through a buffer register. By repeating this, all required data are transmitted to PE7 from the processor element PE 0. In a data-processing circuit, correction processing can be performed for inner sign correction to juxtaposition with eight processor elements PE0-PE7.
Next, the data transfer procedure in the case of performing sign correction by the parallel

processors PE0-PE7 of drawing 18 outside DVD is explained.
In drawing 9 , 208 bytes required for outside sign correction are stored in the same lower address of every 6ROW(s) from the 1st ROW to the 78th ROW of DRAM every 16 bytes as above-mentioned. From the address H'0000 of the 1st ROW of DRAM, 8 bytes of data of H'0013 are repeated from the address H'0010 to the 1st line of a buffer register, this is repeated for 8 bytes of data of H'0003 to the 2nd line of a buffer register, and 8 bytes of data of H'0073 are transmitted to the 8th line of a buffer register from the address H'0070. Thereby, 8 bytes of the beginning from the 1st line of an ECC block to the 8th line were transmitted by the 8th line from the 1st line of a buffer register. This is equivalent also to the first 8 bytes of up to eight trains from [1 of an ECC block train].
Although the C, i.e., train, side was chosen with the R/C terminal, by inner sign correction, by outside sign correction, the R, i.e., line, side is chosen and every 8 bytes of data are transmitted to PE7 from the processor element PE 0 from RR0 to RR7 one by one. The sequential transfer of the data of up to eight trains from [1 of a buffer register train] is carried out from the processor element PE 0 by this actuation at PE7, respectively. since a buffer register becomes empty now -- a degree -- H'0603 from the next address H'0600 of the 7th ROW of DRAM, H'0700 to H'0703, and ... a total of 64 bytes of H'Hfrom 0BF0'0BF3** are transmitted to PE7 from the processor element PE 0 through a buffer register. By repeating this, all required data are transmitted to PE7 from the processor element PE 0.
In a data-processing circuit, correction processing can be performed for outside sign correction to juxtaposition with eight processor elements PE0-PE7.
Although drawing 18 explained carrying out parallel processing of the error correction of DVD to CD-ROM, especially the actuation for DVD is widely applicable to the error correction of ordinary Lead Solomon's product code.
If signal processing to an input circuit is changed according to an input medium and an output circuit or subsequent ones is changed according to an output destination change, this invention can respond also to other media easily. Since the data format of an error correction is similar with the data format of CD-ROM or DVD explained in the above-mentioned example even if it is are recording system media and is broadcast system media, an address-generation circuit, a priority judging circuit, etc. only change some sequence control, and application to an easily different format is attained.
For this reason, it is very advantageous in respect of effective use of a design data.
Drawing 22 is the block diagram of the data processor in which other examples of this invention are shown, and is the case where software realizes the art of this invention.
Although the example shown in drawing 1 constitutes the data processor of this invention from hardware, the address generation in drawing 1 , the access-control circuit 17, the priority judging circuit 16, the data-processing circuit 15, and a buffer register 14 are also realizable with the software of processors, such as a microcontroller.
Drawing 22 shows the hardware configuration on condition of the case where the software on a microcontroller 223 realizes. The input circuit 221 and the output circuit 222 are connected to the microcontroller 223 for main memory 224 and the DRAM interface circuitry 225 through the peripheral bus through the common bus, respectively. The microcontroller 223 is supervising the register (DRAM access request status register Reg st) which requires access, and performs suitable processing according to the access factor over DRAM226 according to the priority of a demand. A DRAM access request status register is a register which shows the existence of a demand to an access factor, it may realize by hardware or software may realize it virtually. There is a register of one or more symbols in an input circuit 221, and if data are prepared, a demand will be come out of and given to a DRAM access request status register. In a microcontroller 223, the error correction routine, the output-processing routine, and the DRAM refresh control routine are operating, and a demand is given to a DRAM access request status register if needed.
Drawing 23 is the flow chart of the priority judging routine of a microprocessor, supervises the DRAM access request status and shows the flow of the software which starts a required processing subroutine suitably.

The contents of the DRAM access request status register are read first (step 230,231), the existence of an access request is investigated according to priority, if there is a demand, the subroutine according to the demand will be started, a demand is reset, and it returns to reading of a status register. The priority shown here is an example to the last, and it is also fully considered that ranking is changed when a system is constituted. Moreover, it may be changed also working [a program].
The refresh processing subroutine 232,240,248 is a subroutine which performs time management with the timer which is the circumference function of a microcontroller 223, and is refreshed for DRAM a fixed period. write from a demodulator circuit As for the subroutine 233,241,249 which performs access, a configuration changes with the locations and approaches of data on the class of medium, or DRAM.
Read-out of read-out for the read-out processing subroutine (step 236,244,252) for the read-out access manipulation routine (step 235,243,251) for read-out access processing subroutine [for C1 correction] (step 234,242,250) and C2 correction and an output and correction, a write-in processing subroutine (step 237,245,253), and others and a write-in processing subroutine (steps 238, 239, and 246,247,254,255) are started according to priority.
Drawing 24 is write from a demodulator circuit. It is the flow chart which shows an example of an access manipulation routine.
Here, in the system which restores to CD-ROM, the subroutine which writes in data so that it may become arrangement of the data shown in drawing 5 using DRAM with a data width of face of 8 bits is shown. The input data from a demodulator circuit shall be set to "DATA", and this two data shall be given as an output "i" of 33 ** counter 119 which showed the location of the data at this time to drawing 3 . However, this 33 ** counter 119 is easily realizable with hardware or software. Since DATA is a sub-code at the time of i= 0 (step 261), a sub-code processing subroutine is started (step 267). Since the sub-code processing itself is essentially unrelated, it abbreviates explanation to this invention. When the number of i is even, one-frame delay is given (step 262,263), and in the case of odd number, it writes in without delay at DRAM (step 262,268). out in a flow buf is an array variable and stores 32 bytes of data written in DRAM next. dly buf is an array variable for one-frame delay, and holds the data of one frame ago (step 265).
When the number of data is odd (step 262), it is dly of one frame ago. out which outputs but [i] to a degree It stores in buf [i] (step 270). Moreover, after carrying out all bit flipping of the parity, it is used for processing as DATA of i= 13, and 14, 15, 16, 29, 30, 31 and 32 is shown to the specification of CD of drawing 20 by parity (step 264,269).
Since 32 bytes is written in continuously, it waits for the next input, without accessing DRAM until it becomes the tail end of i= 32, i.e., a frame. It is out when set to i= 32 (step 266). Since 32 bytes of data which should be written in DRAM will be stored in buf, the address is actually published to DRAM and write-in actuation is performed. The inside of a flow and Variables row, col0, and col are variables for the low of DRAM, an initial column, and column address count, respectively, and are Variable DRAM. ROW, DRAM COL, DRAM DATA is the low, the column address, and data which are actually published by DRAM, respectively.
CD increases every [1] for every frame, and the row address of DRAM returns to 0 with a suitable value (271 reference). Moreover, if the initial column address col0 increases every [32] (it is [every / 20 / H'] at 24 ******) and progresses to the maximum in the same low for every frame, it will return to 0 next (272 reference). When an initial column address is called for, a column address is actually published to DRAM and it writes in coincidence by outputting data to DRAM (step 271,272). Then, a column address is increased by 33 (H'21) every within the same row address, and remaining 31 bytes of writing is performed one by one (step 274).
When a column address tends to exceed maximum H'400, the increment in the write cycle by the recurrence line of a row address is prevented by deducting H'400 and turning up to the young column address in the same row address (274 reference).
The rearrangement circuit of the data realized by the hardware shown in drawing 3 with the software which realizes the flow chart of drawing 24 explained above is realizable with software. Hereafter, although subroutines, such as read-out correction writing of the data for read-out of

the data for read-out of the data for C1 and C2 correction and an output and correction, are required, these are accesses to the data already rearranged appropriately, and since it is easily realizable, explanation is omitted.
When software realizes control of DRAM, it is effective in the ability to use the space where DRAM is excessive as a work area of the data of another purpose.
Availability on industry As stated above, since the data processor of this invention can respond to the error correction of the signal from a medium which can use high-speed access in the same ROW address of DRAM, and is different in common, it becomes unnecessary to use expensive memory for it especially at a high speed, and it can realize a cheap data processor. Furthermore, if the input circuit and output circuit linked to an internal bus are designed so that it may be adapted for a system, since correction processing circuits including DRAM access can be diverted to a system which is different only by some modification, it is effective in development cost being reducible.
[Brief Description of the Drawings]
Drawing 1 is the block diagram of the data processor in which the 1st example of this invention is shown.
Drawing 2 is the detailed block diagram of the input circuit in drawing 1 .
Drawing 3 is the 1st data rearrangement circuit in drawing 1 , and the detailed block diagram of a counter.
Drawing 4 is a timing chart explaining actuation of the input circuit of a CD-ROM signal.
Drawing 5 is drawing showing the example of data allotment on 8-bit [/word] DRAM of CD-ROM data.
Drawing 6 is drawing showing the example of data allotment on 8-bit [/word] DRAM of DVD data.
Drawing 7 is the block diagram showing one example of the 2nd data rearrangement circuit of this invention.
Drawing 8 is drawing showing the example of data allotment on 16-bit [/word] DRAM of CD-ROM data.
Drawing 9 is drawing showing the example of data allotment on 16-bit [/word] DRAM of DVD data.
Drawing 10 is the block diagram showing other examples of the 2nd data rearrangement circuit of this invention.
Drawing 11 is drawing showing the example of data allotment on 32-bit [/word] DRAM of CD-ROM data.
Drawing 12 is drawing showing the example of data allotment on 32-bit [/word] DRAM of DVD data.
Drawing 13 is the block diagram showing the 1st example (for 8-bit [/word] DRAM) of the output circuit in drawing 1 .
Drawing 14 is the block diagram showing the 2nd example (for 16-bit [/word] DRAM) of the output circuit in drawing 1 .
Drawing 15 is the block diagram showing the 3rd example (for 32-bit [/word] DRAM) of the output circuit in drawing 1 .
Drawing 16 is the block diagram showing the example of the address-generation circuit in drawing 1 .
Drawing 17 is the block diagram of the example of CD-ROM / DVD regenerative apparatus which applied this invention.
Drawing 18 is the block diagram showing the example of the data-processing circuit of this invention.
Drawing 19 is the format Fig. of the error correcting code of the time series data after a CD-ROM recovery, and CD-ROM data.
Drawing 20 is drawing showing the error correction of CD-ROM, and the specification of DIN TARIBU.
Drawing 21 is the format Fig. of the error correcting code of the time series data after a CD-ROM recovery, and CD-ROM data.

Drawing 22 is the block diagram of the data processor in which the 2nd example of this invention is shown.
Drawing 23 is the flow chart of the priority judging routine of the microprocessor in drawing 22 .
Drawing 24 is write from the demodulator circuit in drawing 22 . It is the flow chart of an access manipulation routine.

[Translation done.]

図1

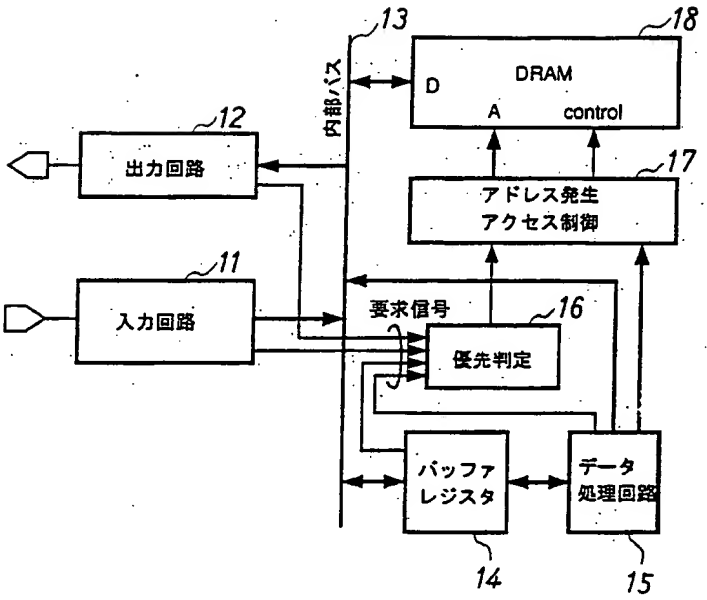

13
18
内部バス
D
DRAM
A
control
12
出力回路
17
アドレス発生
アクセス制御
11
入力回路
要求信号
16
優先判定
バッファ
レジスタ
データ
処理回路
14
15

図2

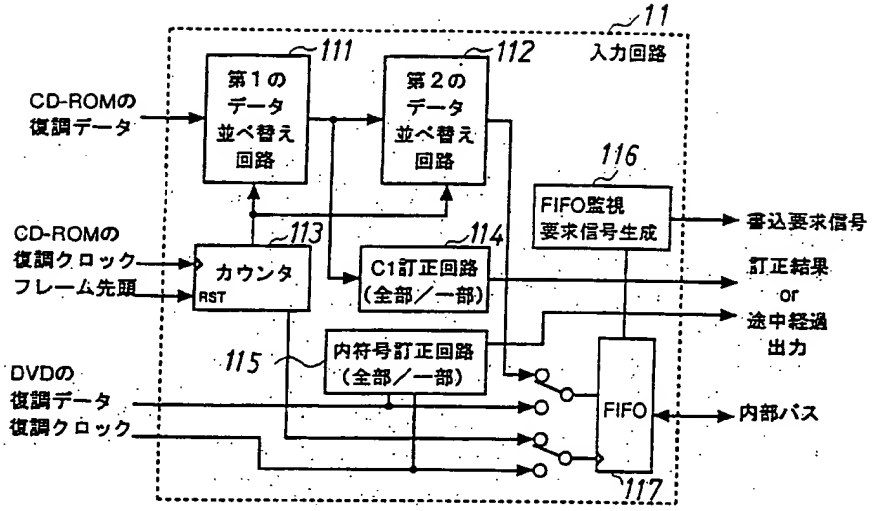

11
入力回路
111
第1の
データ
並べ替え
回路
112
第2の
データ
並べ替え
回路
CD-ROMの
復調データ
116
FIFO監視
要求信号生成
書込要求信号
CD-ROMの
復調クロック
フレーム先頭
113
カウンタ
RST
114
C1訂正回路
(全部／一部)
訂正結果
or
途中経過
出力
115
内符号訂正回路
(全部／一部)
DVDの
復調データ
復調クロック
FIFO
内部バス
117

図3

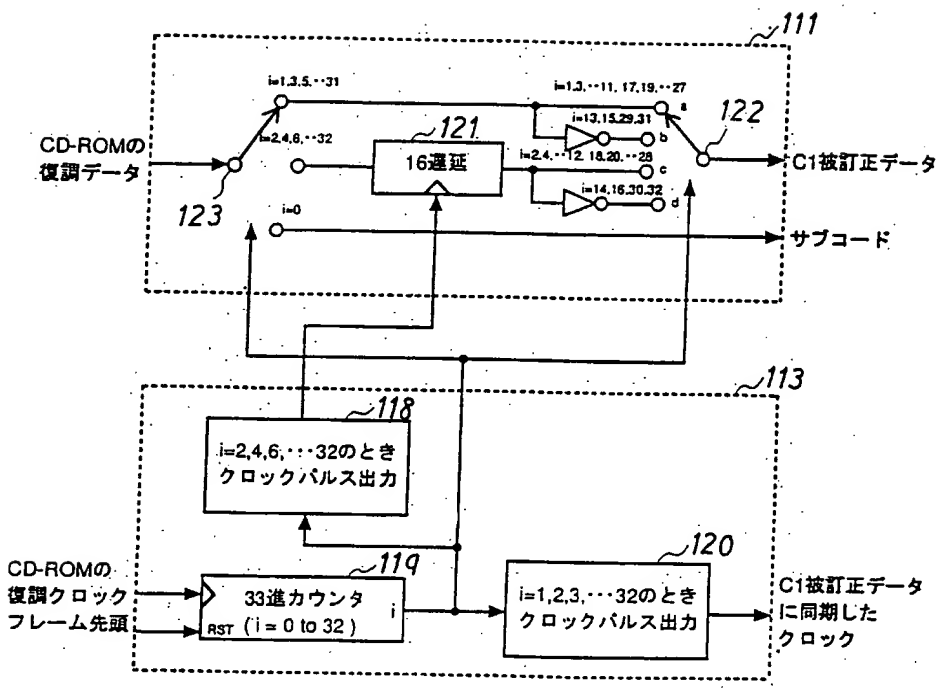

111
i=1,3,5,・・31
i=1,3,・・11, 17,19,・・27
i=13,15,29,31
i=2,4,6,・・32
121
16遅延
122
CD-ROMの
復調データ
123
i=2,4,・・12, 18,20,・・28
i=14,16,30,32
i=0
C1被訂正データ
サブコード
113
118
i=2,4,6,・・・32のとき
クロックパルス出力
120
CD-ROMの
復調クロック
フレーム先頭
119
33進カウンタ
RST (i = 0 to 32)
i
i=1,2,3,・・・32のとき
クロックパルス出力
C1被訂正データ
に同期した
クロック

図4

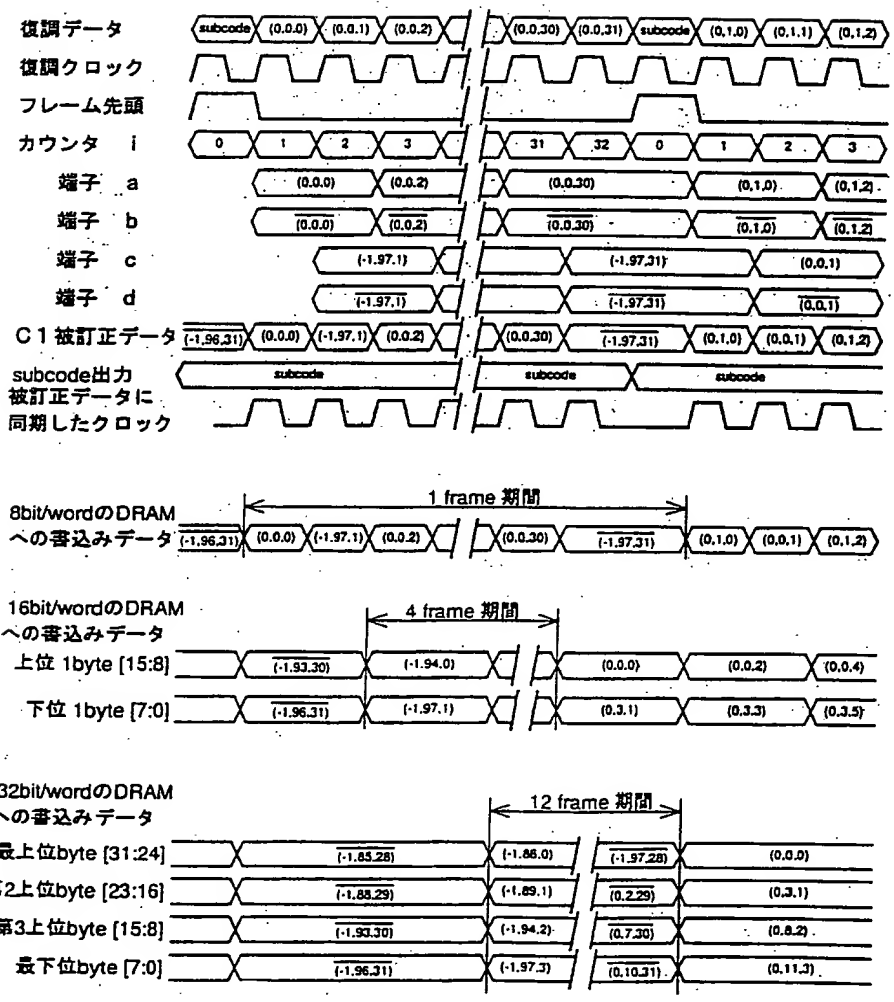
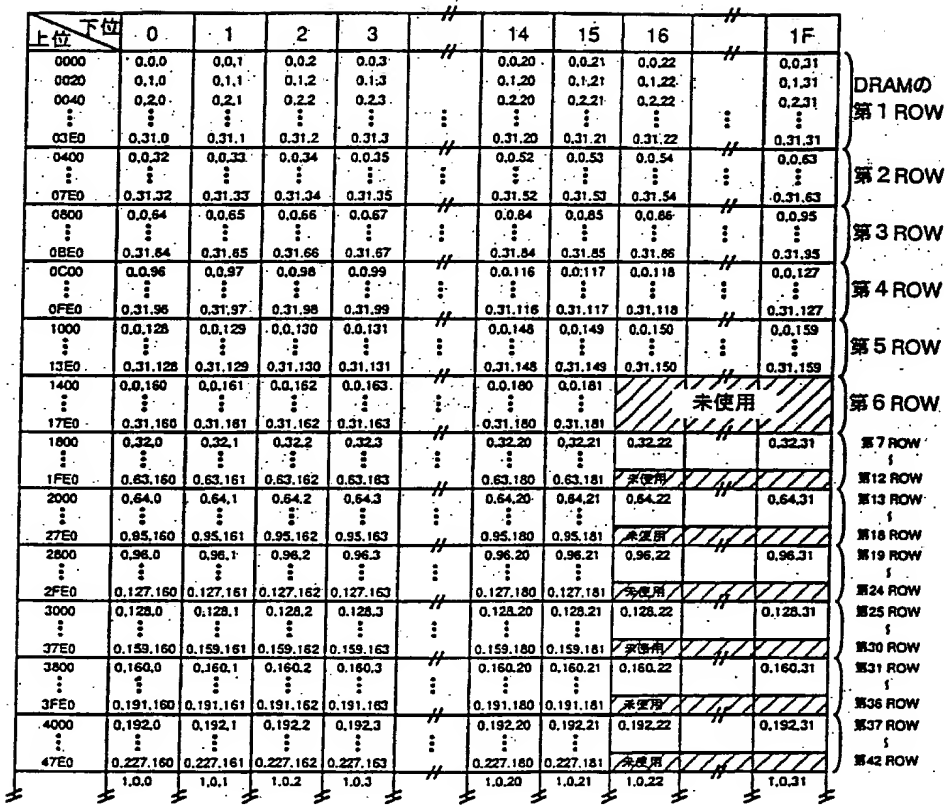

復調データ
復調クロック
フレーム先頭
カウンタ i
端子 a
端子 b
端子 c
端子 d
C1被訂正データ
subcode出力
被訂正データに
同期したクロック
subcode
1 frame 期間
8bit/wordのDRAM
への書込みデータ
16bit/wordのDRAM
への書込みデータ
4 frame 期間
上位 1byte [15:8]
下位 1byte [7:0]
32bit/wordのDRAM
への書込みデータ
12 frame 期間
最上位byte [31:24]
第2上位byte [23:16]
第3上位byte [15:8]
最下位byte [7:0]

図5

上位＼下位	0	1	2	3	4	5	6	7		1A	1B	1C		1F	
000	0.0.0	0.3.1	0.8.2	0.11.3	0.16.4	0.19.5	0.24.6	0.27.7		1.6.26	1.9.27	1.14.28		1.25.31	DRAMØ 第1 ROW
020	-1.94.0	-1.97.3	0.4.2	0.7.3	0.12.4	0.15.5	0.20.6	0.23.7		1.2.26	1.5.27	1.10.28		1.21.31	
040	-1.90.0	-1.93.1	0.0.2	0.3.3	0.8.4	0.11.5	0.16.6	0.19.7		0.96.26	1.1.27	1.6.28		1.17.31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
3E0	0.4.0	0.7.1	0.12.2	0.15.3	0.20.4	0.23.5	0.28.6	0.31.7		1.10.26	1.13.27	1.18.28		-1.97.31	
400	0.1.0	0.4.1	0.9.2	0.12.3	0.17.4	0.20.5	0.25.6	0.28.7		1.7.26	1.10.27	1.15.28		1.26.31	DRAMØ 第2 ROW
420	-1.95.0	0.0.1	0.5.2	0.8.3	0.13.4	0.16.5	0.21.6	0.24.7		1.3.26	1.6.27	1.11.28		1.22.31	
440	-1.91.0	-1.94.1	0.1.2	0.4.3	0.9.4	0.12.5	0.17.6	0.20.7		0.97.26	1.2.27	1.7.28		1.18.31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
7E0	0.5.0	0.8.1	0.13.2	0.16.3	0.21.4	0.24.5	0.27.6	0.32.7		1.11.26	1.14.27	1.19.28		0.0.31	
800	0.2.0	0.5.1	0.10.2	0.13.3	0.18.4	0.21.5	0.26.6	0.29.7		1.8.26	1.11.27	1.16.28		1.27.31	DRAMØ 第3 ROW
820	-1.96.0	0.1.1	0.6.2	0.9.3	0.14.4	0.17.5	0.22.6	0.25.7		1.4.26	1.7.27	1.12.28		1.23.31	
840	-1.92.0	-1.95.1	0.2.2	0.5.3	0.10.4	0.13.5	0.18.6	0.21.7		1.0.26	1.3.27	1.8.28		1.19.31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
BE0	0.6.0	0.9.1	0.14.2	0.17.3	0.22.4	0.25.5	0.30.6	0.33.7		1.12.26	1.15.27	1.20.28		0.1.31	
C00	0.3.0	0.6.1	0.11.2	0.14.3	0.19.4	0.22.5	0.27.6	0.30.7		1.9.26	1.12.27	1.17.28		1.27.31	DRAMØ 第4 ROW
C20	-1.97.0	0.2.1	0.7.2	0.10.3	0.15.4	0.18.5	0.23.6	0.26.7		1.5.26	1.8.27	1.13.28		1.24.31	
C40	-1.93.0	-1.96.1	0.3.2	0.6.3	0.11.4	0.14.5	0.19.6	0.22.7		1.1.26	1.4.27	1.9.28		1.20.31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
FE0	0.7.0	0.10.1	0.15.2	0.18.3	0.23.4	0.26.5	0.31.6	0.34.7		1.13.26	1.16.27	1.21.28		0.2.31	

図6



注）データ（b, 208-227, c）は便宜的にメモリ上のアドレス割付けがなされているが実際は該当するデータがないので割り付けられたメモリ領域は未使用となる。

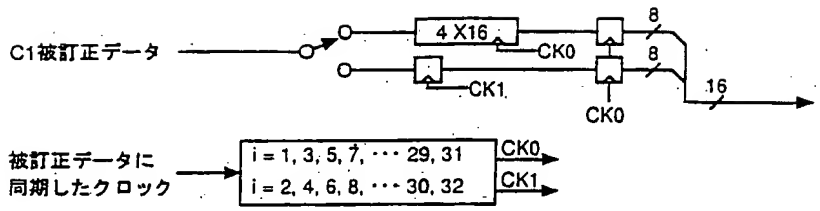

C1被訂正データ
4 X16
CK0
CK1
8
8
16
CK0
被訂正データに
同期したクロック
i = 1, 3, 5, 7, ・・・ 29, 31
i = 2, 4, 6, 8, ・・・ 30, 32
CK0
CK1

図7

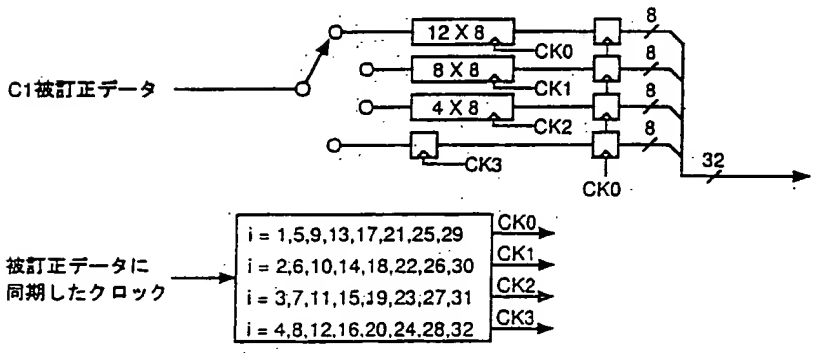
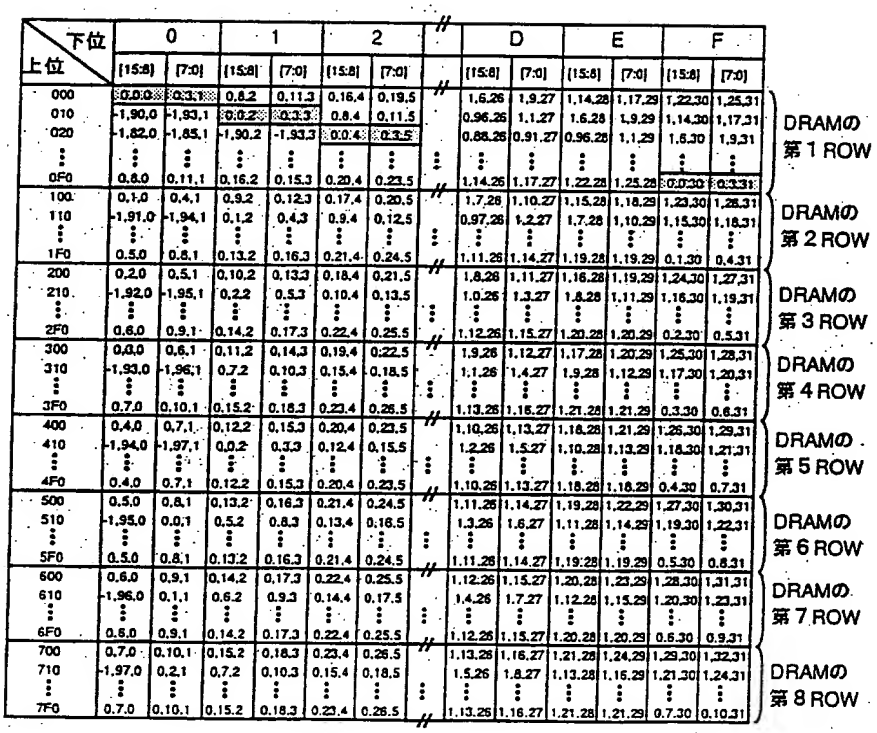

C1被訂正データ
12 X 8
8 X 8
4 X 8
CK0
CK1
CK2
CK3
8
8
8
8
32
CK0
被訂正データに
同期したクロック
i = 1,5,9,13,17,21,25,29
i = 2,6,10,14,18,22,26,30
i = 3,7,11,15,19,23,27,31
i = 4,8,12,16,20,24,28,32
CK0
CK1
CK2
CK3

図10

図8



下位 / 上位	0		1		2		//	D		E		F		
	[15:8]	[7:0]	[15:8]	[7:0]	[15:8]	[7:0]		[15:8]	[7:0]	[15:8]	[7:0]	[15:8]	[7:0]	
000	0,0,0	0,3,1	0,8,2	0,11,3	0,16,4	0,19,5	//	1,6,26	1,9,27	1,14,28	1,17,29	1,22,30	1,25,31	DRAMの第1 ROW
010	-1,90,0	-1,93,1	0,0,2	0,3,3	0,8,4	0,11,5		0,96,26	1,1,27	1,6,28	1,9,29	1,14,30	1,17,31	
020	-1,82,0	-1,85,1	-1,90,2	-1,93,3	0,0,4	0,3,5		0,86,26	0,91,27	0,96,28	1,1,29	1,6,30	1,9,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
0F0	0,8,0	0,11,1	0,16,2	0,19,3	0,20,4	0,23,5	//	1,14,26	1,17,27	1,22,28	1,25,28	0,0,30	0,3,31	
100	0,1,0	0,4,1	0,9,2	0,12,3	0,17,4	0,20,5	//	1,7,26	1,10,27	1,15,28	1,18,29	1,23,30	1,26,31	DRAMの第2 ROW
110	-1,91,0	-1,94,1	0,1,2	0,4,3	0,9,4	0,12,5		0,97,26	1,2,27	1,7,28	1,10,29	1,15,30	1,18,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
1F0	0,5,0	0,8,1	0,13,2	0,16,3	0,21,4	0,24,5	//	1,11,26	1,14,27	1,19,28	1,19,29	0,1,30	0,4,31	
200	0,2,0	0,5,1	0,10,2	0,13,3	0,18,4	0,21,5	//	1,8,26	1,11,27	1,16,28	1,19,29	1,24,30	1,27,31	DRAMの第3 ROW
210	-1,92,0	-1,95,1	0,2,2	0,5,3	0,10,4	0,13,5		1,0,26	1,3,27	1,8,28	1,11,29	1,16,30	1,19,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
2F0	0,6,0	0,9,1	0,14,2	0,17,3	0,22,4	0,25,5	//	1,12,26	1,15,27	1,20,28	1,20,29	0,2,30	0,5,31	
300	0,0,0	0,6,1	0,11,2	0,14,3	0,19,4	0,22,5	//	1,9,26	1,12,27	1,17,28	1,20,29	1,25,30	1,28,31	DRAMの第4 ROW
310	-1,93,0	-1,96,1	0,7,2	0,10,3	0,15,4	0,18,5		1,1,26	1,4,27	1,9,28	1,12,29	1,17,30	1,20,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
3F0	0,7,0	0,10,1	0,15,2	0,18,3	0,23,4	0,26,5	//	1,13,26	1,16,27	1,21,28	1,21,29	0,3,30	0,6,31	
400	0,4,0	0,7,1	0,12,2	0,15,3	0,20,4	0,23,5	//	1,10,26	1,13,27	1,18,28	1,21,29	1,26,30	1,29,31	DRAMの第5 ROW
410	-1,94,0	-1,97,1	0,0,2	0,3,3	0,12,4	0,15,5		1,2,26	1,5,27	1,10,28	1,13,29	1,18,30	1,21,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
4F0	0,4,0	0,7,1	0,12,2	0,15,3	0,20,4	0,23,5	//	1,10,26	1,13,27	1,18,28	1,18,29	0,4,30	0,7,31	
500	0,5,0	0,8,1	0,13,2	0,16,3	0,21,4	0,24,5	//	1,11,26	1,14,27	1,19,28	1,22,29	1,27,30	1,30,31	DRAMの第6 ROW
510	-1,95,0	0,0,1	0,5,2	0,8,3	0,13,4	0,16,5		1,3,26	1,6,27	1,11,28	1,14,29	1,19,30	1,22,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
5F0	0,5,0	0,8,1	0,13,2	0,16,3	0,21,4	0,24,5	//	1,11,26	1,14,27	1,19,28	1,19,29	0,5,30	0,8,31	
600	0,6,0	0,9,1	0,14,2	0,17,3	0,22,4	0,25,5	//	1,12,26	1,15,27	1,20,28	1,23,29	1,28,30	1,31,31	DRAMの第7 ROW
610	-1,96,0	0,1,1	0,6,2	0,9,3	0,14,4	0,17,5		1,4,26	1,7,27	1,12,28	1,15,29	1,20,30	1,23,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
6F0	0,6,0	0,9,1	0,14,2	0,17,3	0,22,4	0,25,5	//	1,12,26	1,15,27	1,20,28	1,20,29	0,6,30	0,9,31	
700	0,7,0	0,10,1	0,15,2	0,18,3	0,23,4	0,26,5	//	1,13,26	1,16,27	1,21,28	1,24,29	1,29,30	1,32,31	DRAMの第8 ROW
710	-1,97,0	0,2,1	0,7,2	0,10,3	0,15,4	0,18,5		1,5,26	1,8,27	1,13,28	1,16,29	1,21,30	1,24,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
7F0	0,7,0	0,10,1	0,15,2	0,18,3	0,23,4	0,26,5	//	1,13,26	1,16,27	1,21,28	1,21,29	0,7,30	0,10,31	

図9

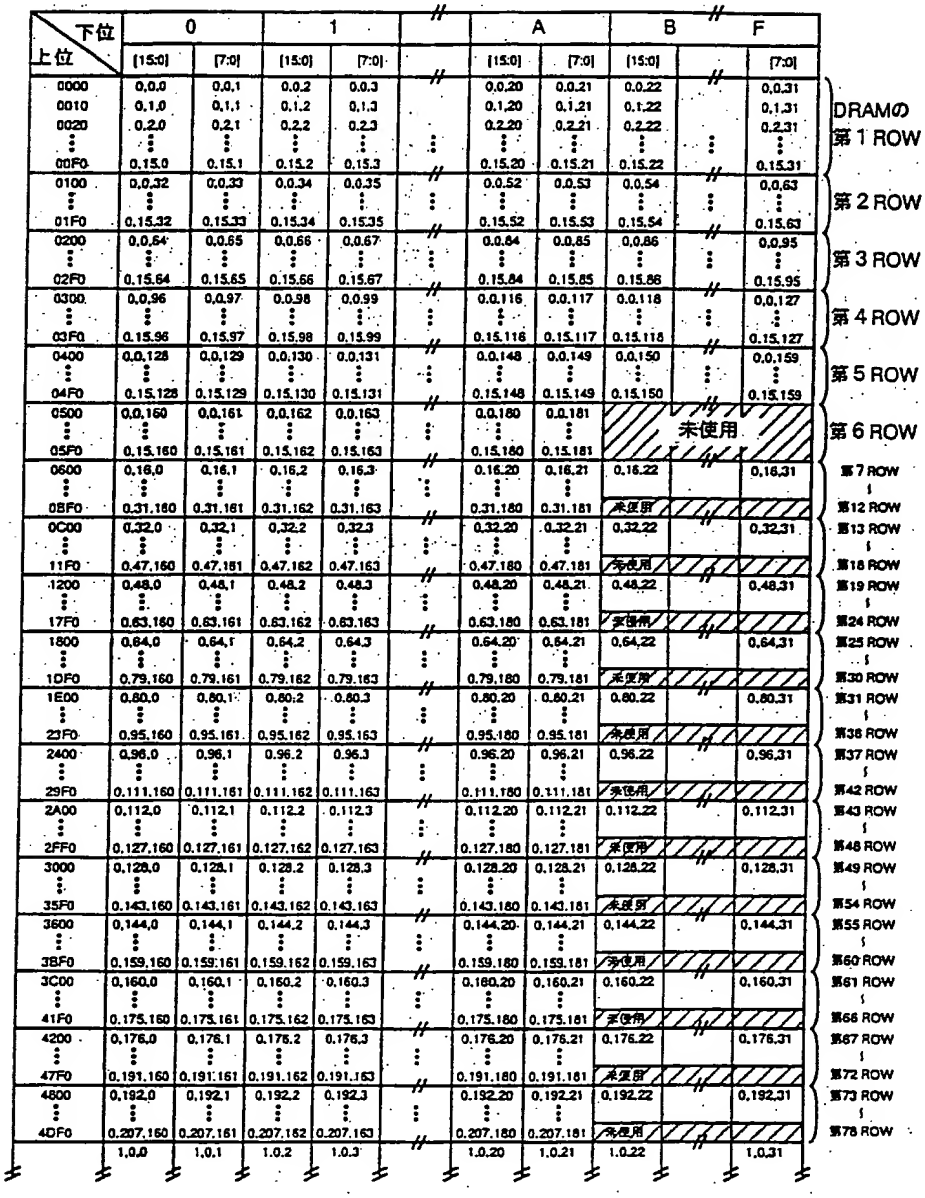

下位
上位
0
1
A
B
F
[15:0]
[7:0]
DRAMの
第1 ROW
第2 ROW
第3 ROW
第4 ROW
第5 ROW
第6 ROW
第7 ROW
第12 ROW
第13 ROW
第18 ROW
第19 ROW
第24 ROW
第25 ROW
第30 ROW
第31 ROW
第36 ROW
第37 ROW
第42 ROW
第43 ROW
第48 ROW
第49 ROW
第54 ROW
第55 ROW
第60 ROW
第61 ROW
第66 ROW
第67 ROW
第72 ROW
第73 ROW
第78 ROW
未使用

図１１

上位＼下位	0				1			6		7				
	[31:24]	[23:16]	[15:8]	[7:0]	[31:24]	[23:16]		[15:8]	[7:0]	[31:24]	[23:16]	[15:8]	[7:0]	
000	0.0.0	0.3.1	0.6.2	0.11.3	0.16.4	0.19.5		1.6.26	1.9.27	1.14.28	1.17.29	1.22.30	1.25.31	DRAMの第１ROW
008	-1,62.0	-1,65,1	-1,90.2	-1,93,3	0.0.4	0.3.5		0.88.26	0.91.27	0.96.28	1.1.29	1.6.30	1.9.31	
010	-1,66.0	-1,69,1	-1,74,2	-1,77,3	-1,-62.4	-1,65.5		0,72.26	0,75,27	0.80.28	0.83.29	0.88.30	0.91.31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
038	0.16.0	0.19.1	0.24.2	0.27.3	0.32.4	0.35.5		-1.90.26	-1.93.27	0.0.28	0.3.28	0.8.30	0.11.31	
040	0,1,0	0,4,1	0,9,2	0,12,3	0,17,4	0,20,5		1,7,26	1,10,27	1,15,28	1,18,29	1,23,30	1,26,31	DRAMの第２ROW
048	-1,83,0	-1,86,1	-1,91,2	-1,94,3	-0,1,4	0,4,5		0,89,26	0,92,27	0,97,28	1,2,29	1,7,30	1,10,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
078	0.17.0	0.20.1	0.25.2	0.28.3	0.33.4	0.36.5		-1,91,26	-1,94,27	0.1.28	0.4.29	0.9.30	0.12.31	
080	0,2,0	0,5,1	0,10,2	0,13,3	0,18,4	0,21,5		1,8,26	1,11,27	1,16,28	1,19,29	1,24,30	1,27,31	DRAMの第３ROW
088	-1,84,0	-1,87,1	-1,92,2	-1,95,3	0,2,4	0,5,5		0,90,26	0,93,27	1,0,28	1,3,29	1,8,30	1,11,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
088	0.18.0	0.21.1	0.26.2	0.29.3	0.34.4	0.37.5		-1,92,26	-1,95,27	0.2.28	0.5.29	0.10.30	0.13.31	
0C0	0,3,0	0,6,1	0,11,2	0,14,3	0,19,4	0,22,5		1,9,26	1,12,27	1,17,28	1,20,29	1,25,30	1,28,31	DRAMの第４ROW
0C8	-1,85,0	-1,88,1	-1,93,2	-1,96,3	0,3,4	0,6,5		0,91,26	0,94,27	1,1,28	1,4,29	1,9,30	1,12,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
0F8	0.19.0	0.22.1	0.27.2	0.30.3	0.35.4	0.38.5		-1,93,26	-1,96,27	0.3.28	0.6.29	0.11.30	0.14.31	
100	0,4,0	0,7,1	0,12,2	0,15,3	0,20,4	0,23,5		1,10,26	1,13,27	1,18,28	1,21,29	1,26,30	1,29,31	DRAMの第５ROW
108	-1,86,0	-1,89,1	-1,94,2	-1,97,3	0,4,4	0,7,5		0,92,26	0,95,27	1,2,28	1,5,29	1,10,30	1,13,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
138	0.20.0	0.23.1	0.28.2	0.31.3	0.36.4	0.39.5		-1,94,26	-1,97,27	0.4.28	0.7.29	0.12.30	0.15.31	
140	0,5,0	0,8,1	0,13,2	0,16,3	0,21,4	0,24,5		1,11,26	1,14,27	1,19,28	1,22,29	1,27,30	1,30,31	DRAMの第６ROW
148	-1,87,0	-1,90,1	-1,95,2	0,0,3	0,5,4	0,8,5		0,93,26	0,96,27	1,3,28	1,6,29	1,11,30	1,14,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
178	0.21.0	0.24.1	0.29.2	0.32.3	0.37.4	0.40.5		-1,95,26	0.0.27	0.5.28	0.8.29	0.13.30	0.16.31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
3C0	0,15,0	0,18,1	0,23,2	0,26,3	0,31,4	0,34,5		1,21,26	1,24,27	1,29,28	1,32,29	1,37,30	1,40,31	DRAMの第16ROW
3C8	-1,97,0	0,2,1	0,7,2	0,10,3	0,15,4	0,18,5		1,5,26	1,8,27	1,13,28	1,16,29	1,21,30	1,24,31	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
3F8	0.31.0	0.34.1	0.39.2	0.42.3	0.47.4	0.50.5		0.7.26	0.10.27	0.15.28	0.18.29	0.23.30	0.26.31	

図１２

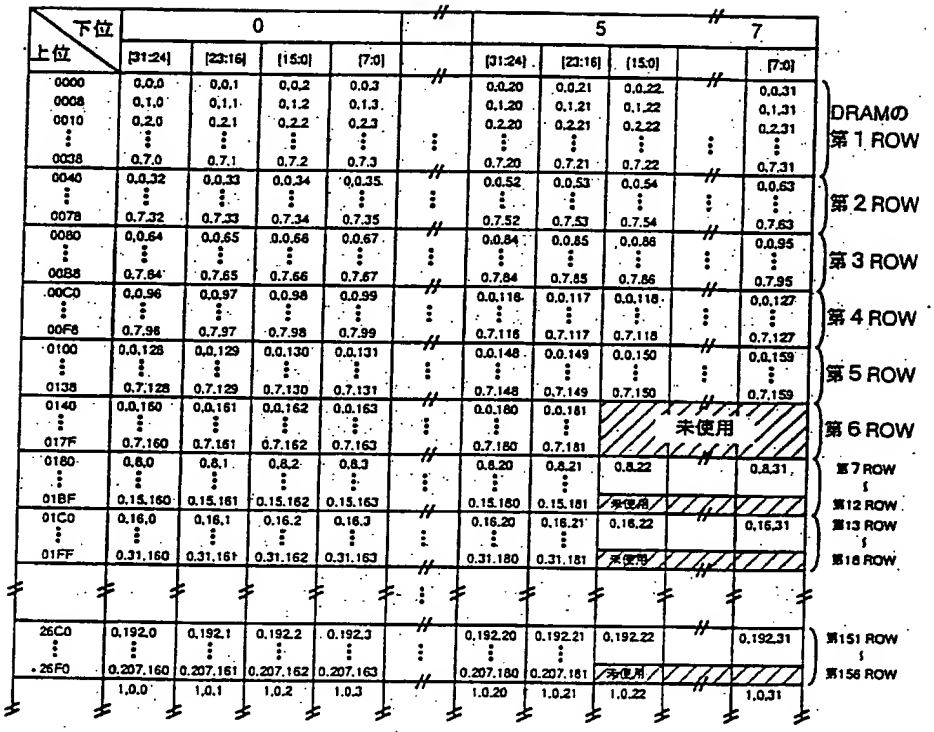

下位
上位
0
5
7
[31:24]
[23:16]
[15:0]
[7:0]
0000
0008
0010
0038
0040
0078
0080
00B8
00C0
00F8
0100
0138
0140
017F
0180
01BF
01C0
01FF
26C0
26F0
0,0,0
0,0,1
0,0,2
0,0,3
0,1,0
0,1,1
0,1,2
0,1,3
0,2,0
0,2,1
0,2,2
0,2,3
0,7,0
0,7,1
0,7,2
0,7,3
0,0,20
0,0,21
0,0,22
0,0,31
0,1,20
0,1,21
0,1,22
0,1,31
0,2,20
0,2,21
0,2,22
0,2,31
0,7,20
0,7,21
0,7,22
0,7,31
0,0,32
0,0,33
0,0,34
0,0,35
0,7,32
0,7,33
0,7,34
0,7,35
0,0,52
0,0,53
0,0,54
0,0,63
0,7,52
0,7,53
0,7,54
0,7,63
0,0,64
0,0,65
0,0,66
0,0,67
0,7,64
0,7,65
0,7,66
0,7,67
0,0,84
0,0,85
0,0,86
0,0,95
0,7,84
0,7,85
0,7,86
0,7,95
0,0,96
0,0,97
0,0,98
0,0,99
0,7,96
0,7,97
0,7,98
0,7,99
0,0,116
0,0,117
0,0,118
0,0,127
0,7,116
0,7,117
0,7,118
0,7,127
0,0,128
0,0,129
0,0,130
0,0,131
0,7,128
0,7,129
0,7,130
0,7,131
0,0,148
0,0,149
0,0,150
0,0,159
0,7,148
0,7,149
0,7,150
0,7,159
0,0,160
0,0,161
0,0,162
0,0,163
0,7,160
0,7,161
0,7,162
0,7,163
0,0,180
0,0,181
未使用
0,7,180
0,7,181
0,8,0
0,8,1
0,8,2
0,8,3
0,8,20
0,8,21
0,8,22
0,8,31
0,15,160
0,15,161
0,15,162
0,15,163
0,15,180
0,15,181
0,16,0
0,16,1
0,16,2
0,16,3
0,16,20
0,16,21
0,16,22
0,16,31
0,31,160
0,31,161
0,31,162
0,31,163
0,31,180
0,31,181
0,192,0
0,192,1
0,192,2
0,192,3
0,192,20
0,192,21
0,192,22
0,192,31
0,207,160
0,207,161
0,207,162
0,207,163
0,207,180
0,207,181
1,0,0
1,0,1
1,0,2
1,0,3
1,0,20
1,0,21
1,0,22
1,0,31
DRAMの
第１ROW
第２ROW
第３ROW
第４ROW
第５ROW
第６ROW
第７ROW
～
第12 ROW
第13 ROW
～
第18 ROW
第151 ROW
～
第156 ROW

図１３

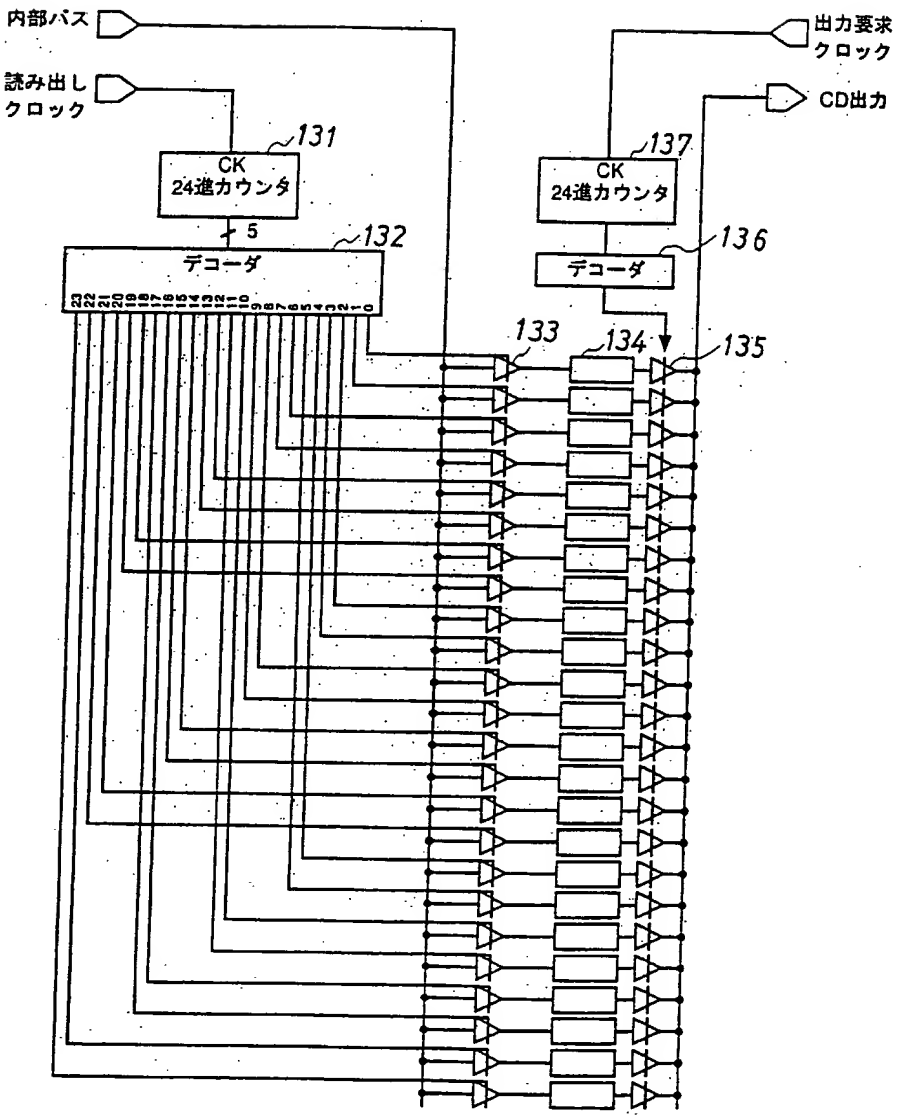

内部バス
読み出し
クロック
CK
24進カウンタ
131
5
デコーダ
132
出力要求
クロック
CD出力
CK
24進カウンタ
137
デコーダ
136
133
134
135

図１４

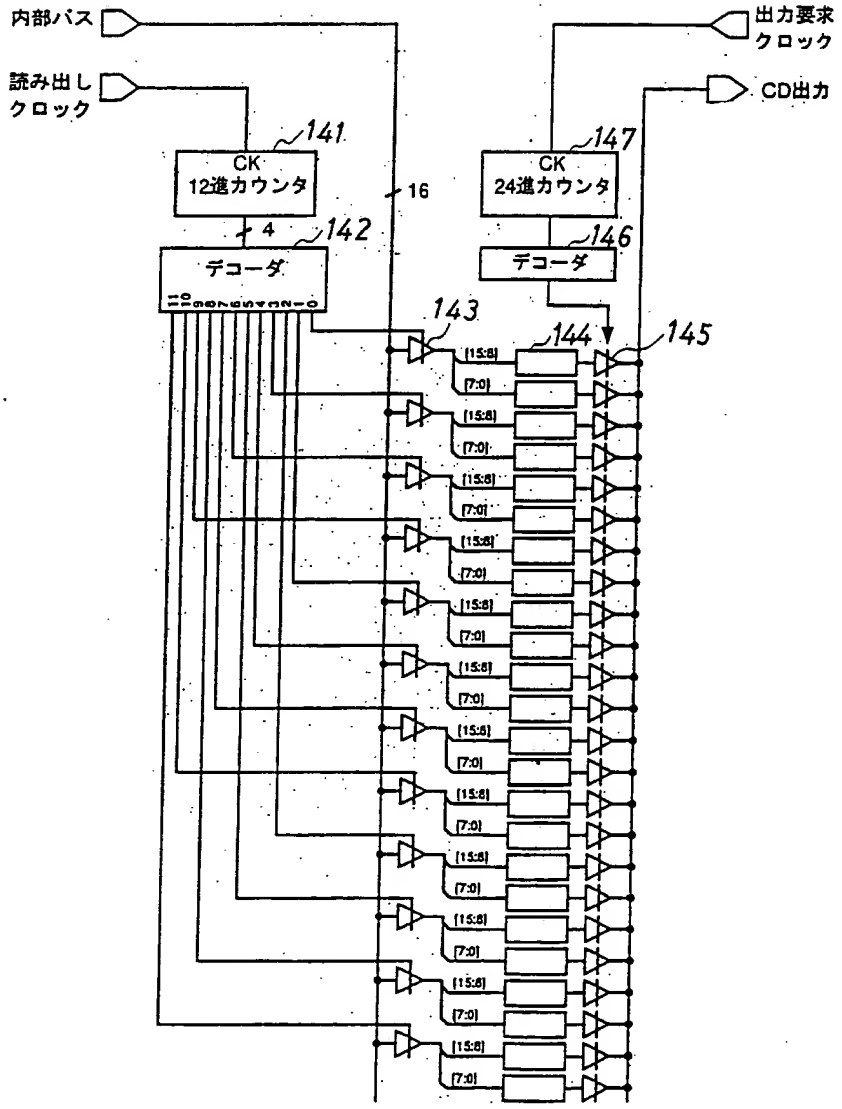

内部バス
読み出し
クロック
出力要求
クロック
CD出力
141
CK
12進カウンタ
16
147
CK
24進カウンタ
4
142
デコーダ
146
デコーダ
143
144
145
[15:8]
[7:0]

図１５

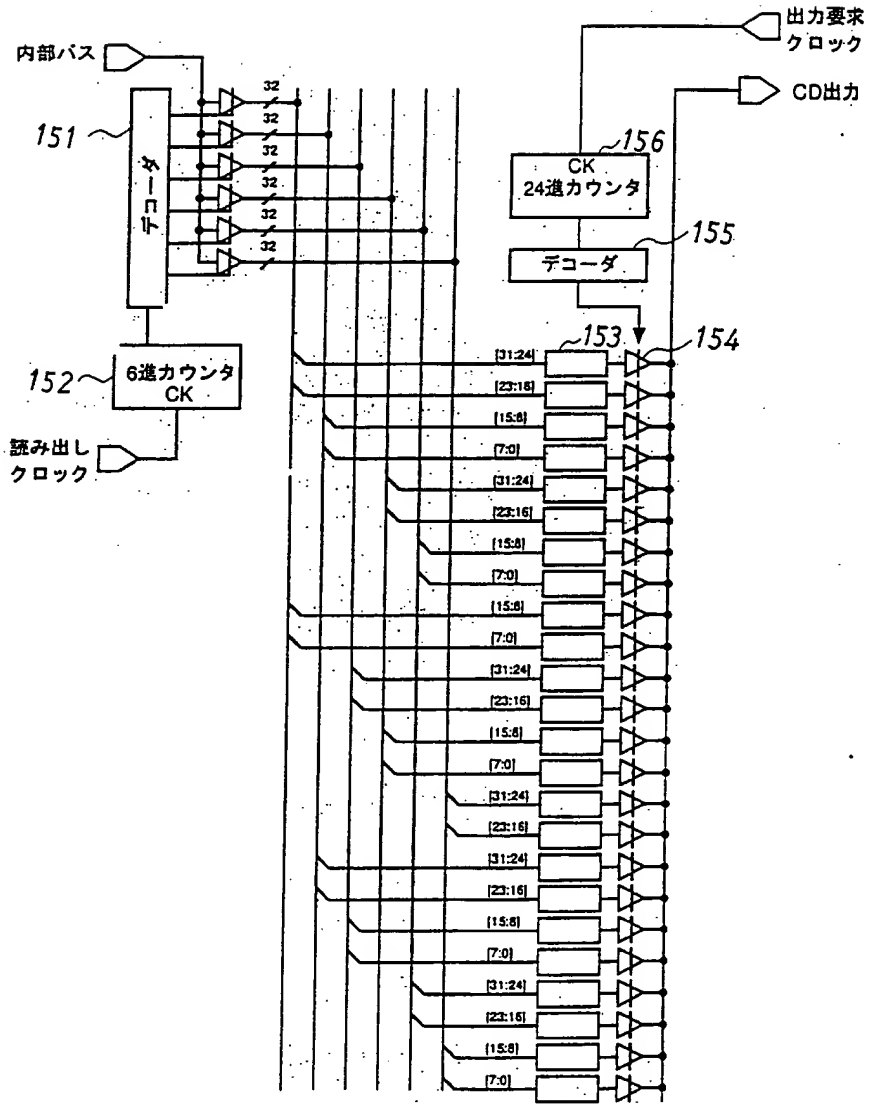

内部バス
出力要求
クロック
CD出力
151
ターミナ
32
156
CK
24進カウンタ
155
デコーダ
153
154
152
6進カウンタ
CK
読み出し
クロック
[31:24]
[23:16]
[15:8]
[7:0]

図１６

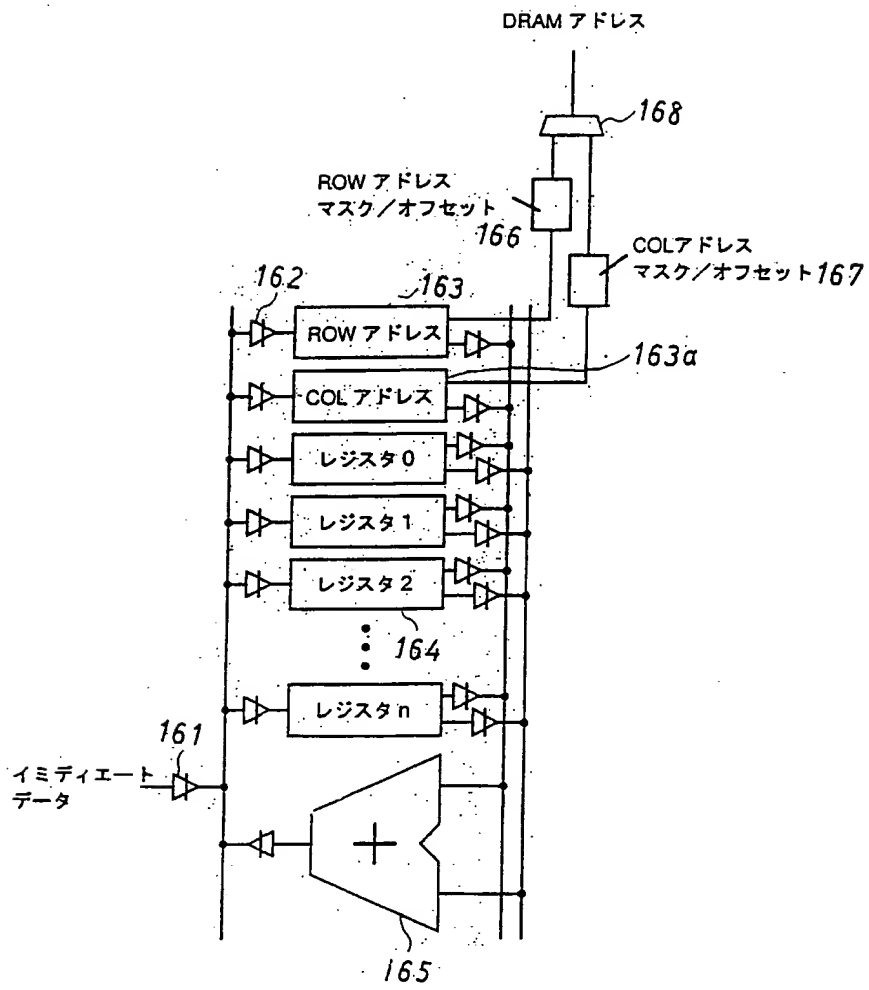

DRAM アドレス
168
ROW アドレス
マスク／オフセット
166
COLアドレス
マスク／オフセット167
162
163
ROW アドレス
163a
COL アドレス
レジスタ 0
レジスタ 1
レジスタ 2
164
レジスタ n
161
イミディエート
データ
165

図１７

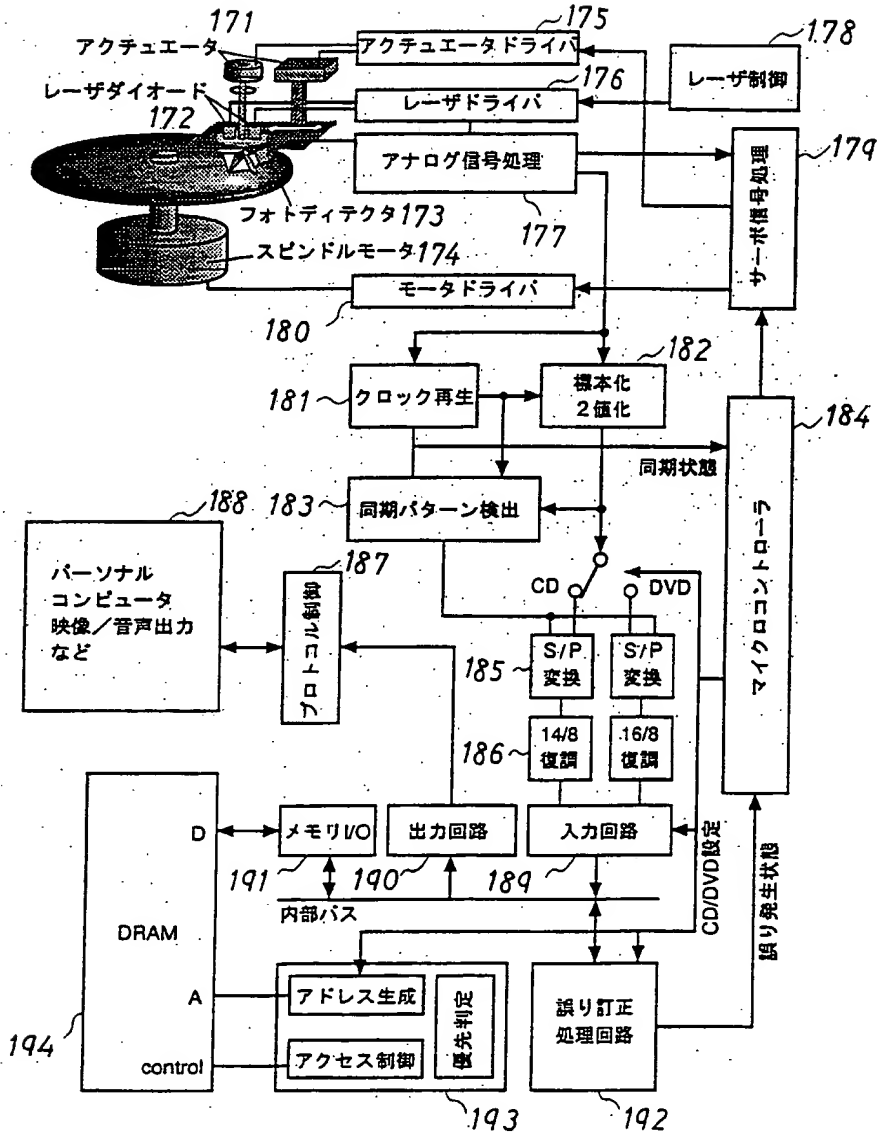

171
アクチュエータ
レーザダイオード
172
フォトディテクタ173
スピンドルモータ174
175
アクチュエータドライバ
176
レーザドライバ
177
アナログ信号処理
178
レーザ制御
179
サーボ信号処理
180
モータドライバ
181
クロック再生
182
標本化
２値化
183
同期パターン検出
184
マイクロコントローラ
同期状態
188
パーソナル
コンピュータ
映像／音声出力
など
187
プロトコル制御
CD
DVD
185
S/P
変換
S/P
変換
186
14/8
復調
16/8
復調
D
メモリI/O
出力回路
入力回路
191
190
189
内部バス
CD/DVD設定
誤り発生状態
DRAM
A
control
アドレス生成
アクセス制御
優先判定
誤り訂正
処理回路
193
192
194

図１８

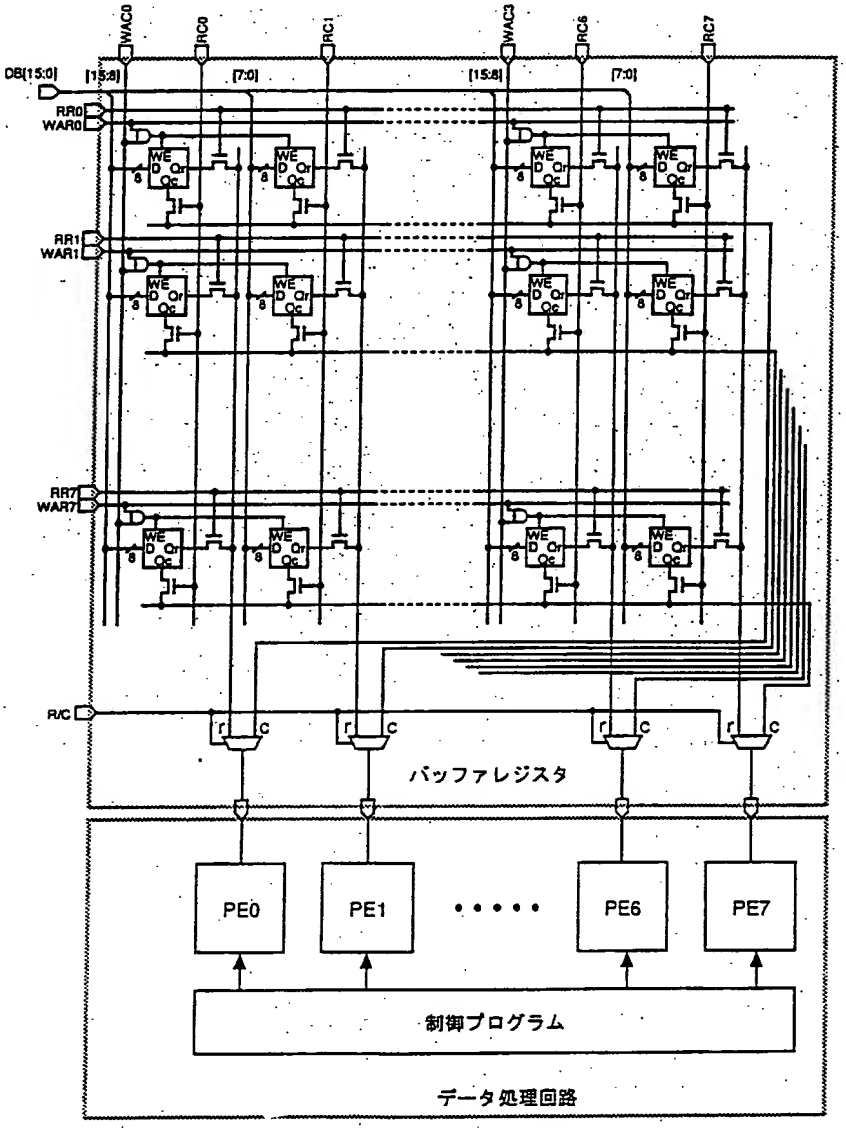
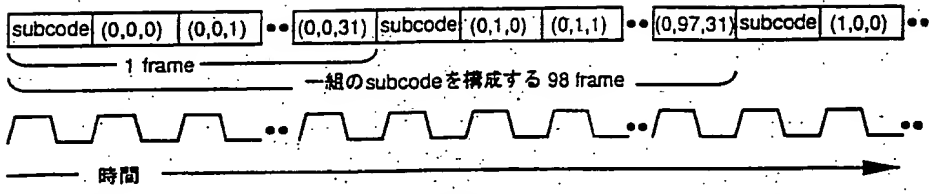

WAC0
RC0
RC1
WAC3
RC6
RC7
DB[15:0]
[15:8]
[7:0]
[15:8]
[7:0]
RR0
WAR0
RR1
WAR1
RR7
WAR7
R/C
バッファレジスタ
PE0
PE1
PE6
PE7
制御プログラム
データ処理回路

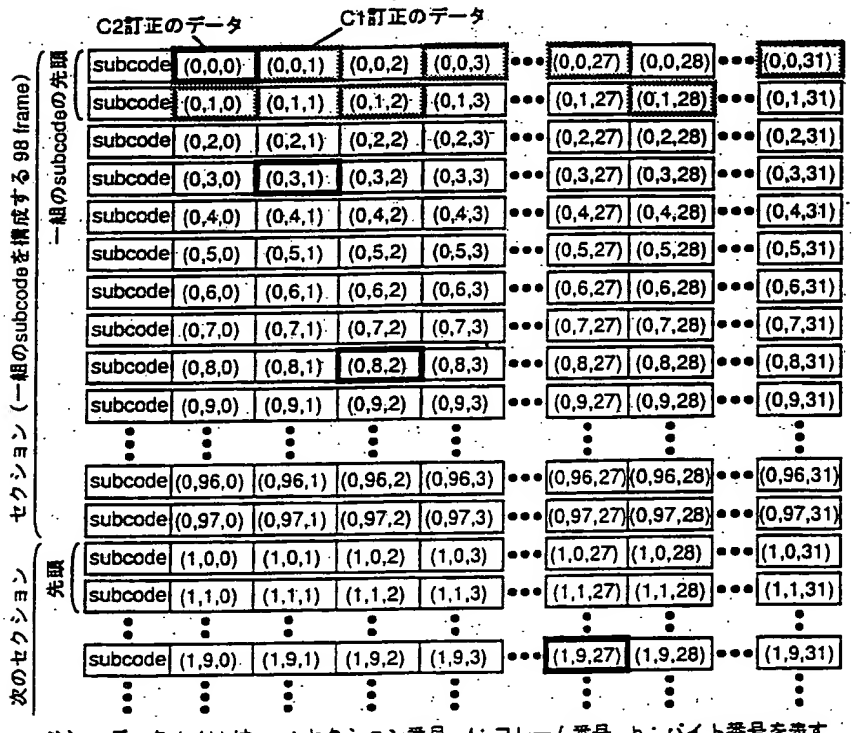
図１９

(a)



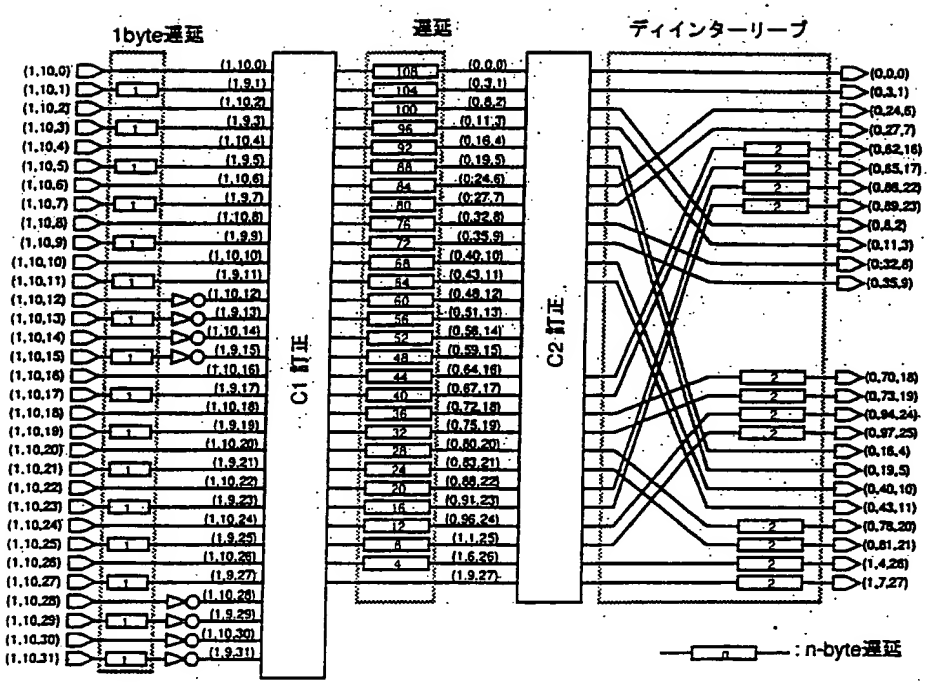
注) データ (s,f,b) は、s：セクション番号、f：フレーム番号、b：バイト番号を表す。

(b)



注) データ (s,f,b) は、s：セクション番号、f：フレーム番号、b：バイト番号を表す。

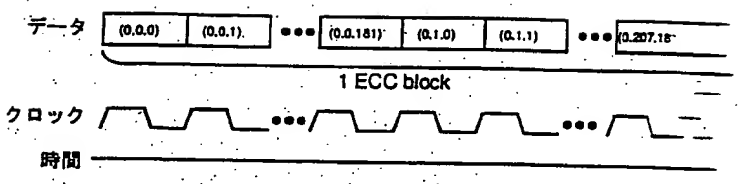
図２０



注） データ (s,f,b) は、s：セクション番号、f：フレーム番号、b：バイト番号を表す。

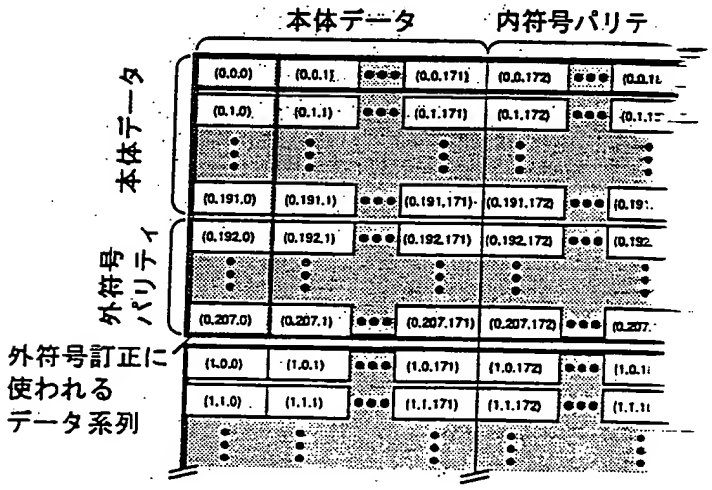
図２１

(a)



注） データ (b,r,c) は、b：ブロック番号、r：行番号、c：列番号

(b)



注） データ (b,r,c) は、b：ブロック番号、r：行番号、c：列番号

図２２

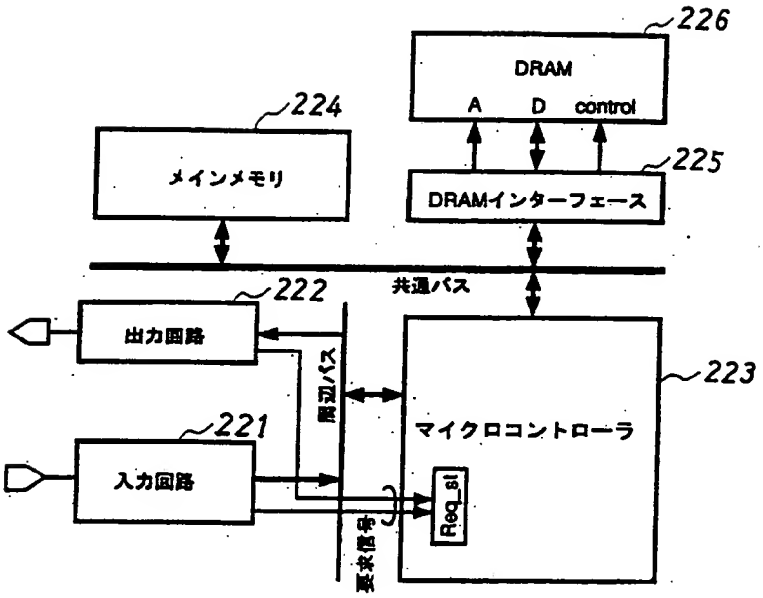

226
DRAM
A
D
control
224
メインメモリ
225
DRAMインターフェース
共通バス
222
出力回路
周辺バス
223
マイクロコントローラ
221
入力回路
Req_st
要求信号

図２３

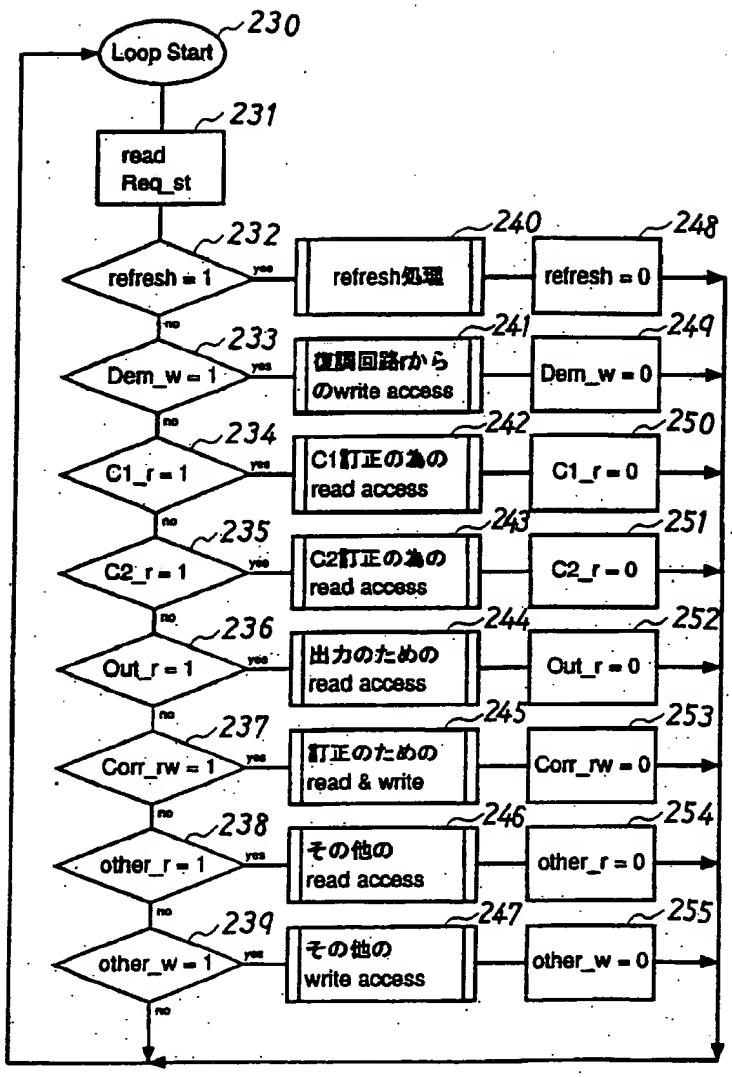

230
Loop Start
231
read
Req_st
232
refresh = 1
yes
240
refresh処理
248
refresh = 0
no
233
Dem_w = 1
yes
241
復調回路rから
のwrite access
249
Dem_w = 0
no
234
C1_r = 1
yes
242
C1訂正の為の
read access
250
C1_r = 0
no
235
C2_r = 1
yes
243
C2訂正の為の
read access
251
C2_r = 0
no
236
Out_r = 1
yes
244
出力のための
read access
252
Out_r = 0
no
237
Corr_rw = 1
yes
245
訂正のための
read & write
253
Corr_rw = 0
no
238
other_r = 1
yes
246
その他の
read access
254
other_r = 0
no
239
other_w = 1
yes
247
その他の
write access
255
other_w = 0
no

図２４